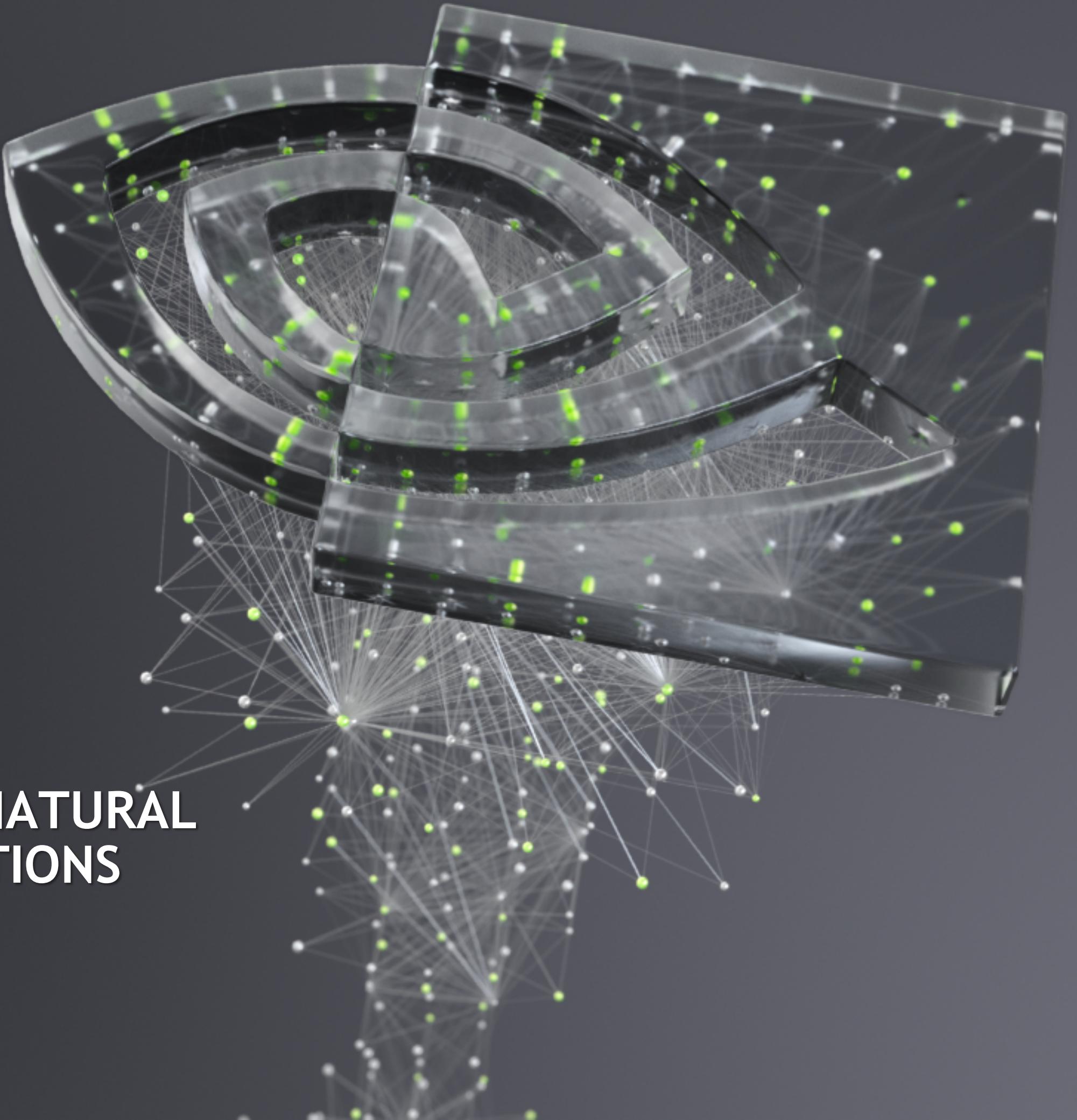




DEEP  
LEARNING  
INSTITUTE

# BUILDING TRANSFORMER-BASED NATURAL LANGUAGE PROCESSING APPLICATIONS

PD. Dr. Juan J. Durillo





Register: [courses.nvidia.com/join](https://courses.nvidia.com/join)

Enroll: [courses.nvidia.com/dli-event](https://courses.nvidia.com/dli-event)

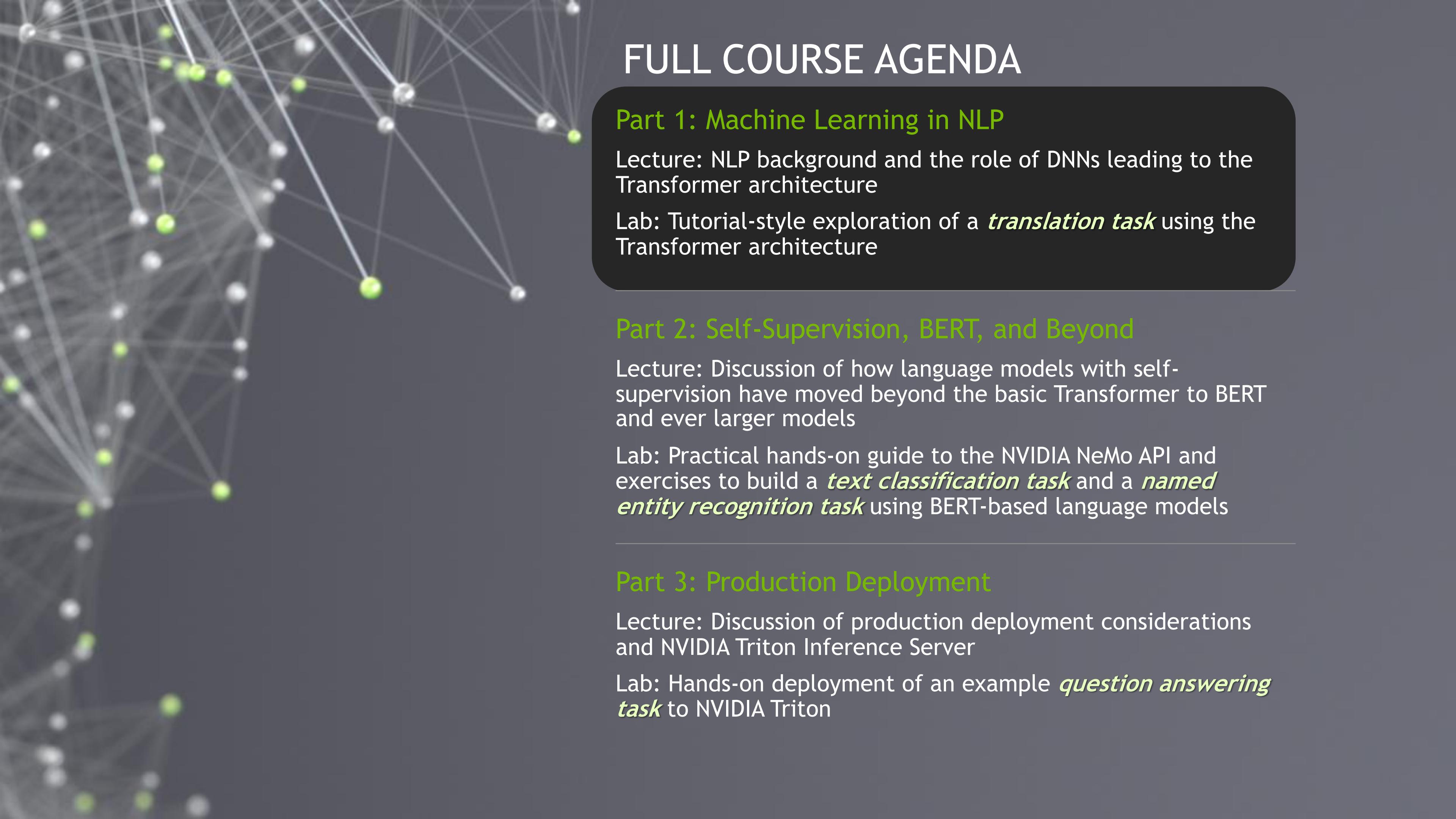
**LRZ\_NLP\_AMBASSADOR\_JY22**

## Disclaimer

After this course you are not an NLP expert

This course bring you no close to a PhD level on NLP

This course might just be the beginning of your NLP journey



# FULL COURSE AGENDA

## Part 1: Machine Learning in NLP

Lecture: NLP background and the role of DNNs leading to the Transformer architecture

Lab: Tutorial-style exploration of a *translation task* using the Transformer architecture

## Part 2: Self-Supervision, BERT, and Beyond

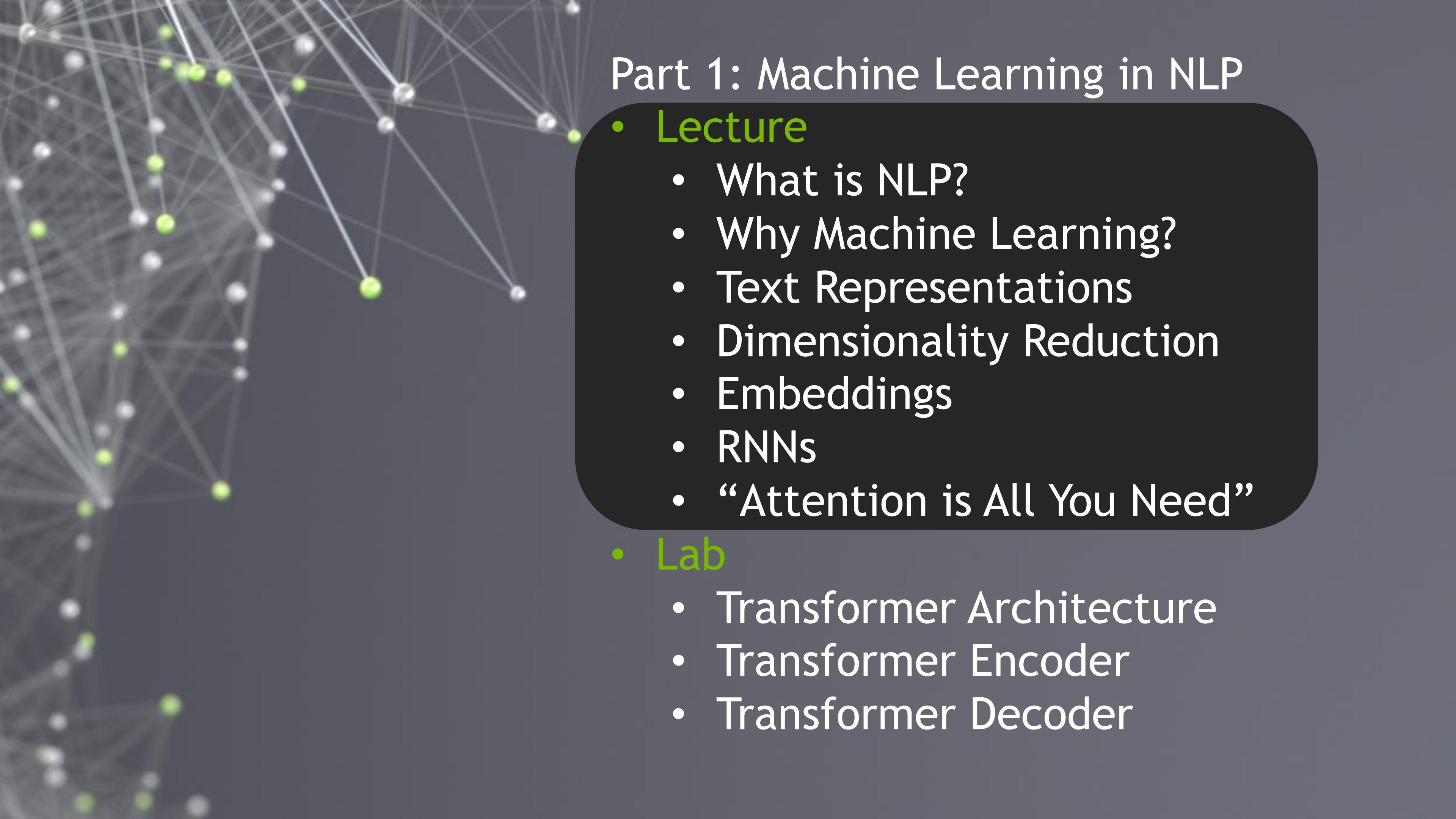
Lecture: Discussion of how language models with self-supervision have moved beyond the basic Transformer to BERT and ever larger models

Lab: Practical hands-on guide to the NVIDIA NeMo API and exercises to build a *text classification task* and a *named entity recognition task* using BERT-based language models

## Part 3: Production Deployment

Lecture: Discussion of production deployment considerations and NVIDIA Triton Inference Server

Lab: Hands-on deployment of an example *question answering task* to NVIDIA Triton



# Part 1: Machine Learning in NLP

- **Lecture**
  - What is NLP?
  - Why Machine Learning?
  - Text Representations
  - Dimensionality Reduction
  - Embeddings
  - RNNs
  - “Attention is All You Need”
- **Lab**
  - Transformer Architecture
  - Transformer Encoder
  - Transformer Decoder

# CONVERSATIONAL AI TECHNOLOGIES



Automatic Speech  
Recognition

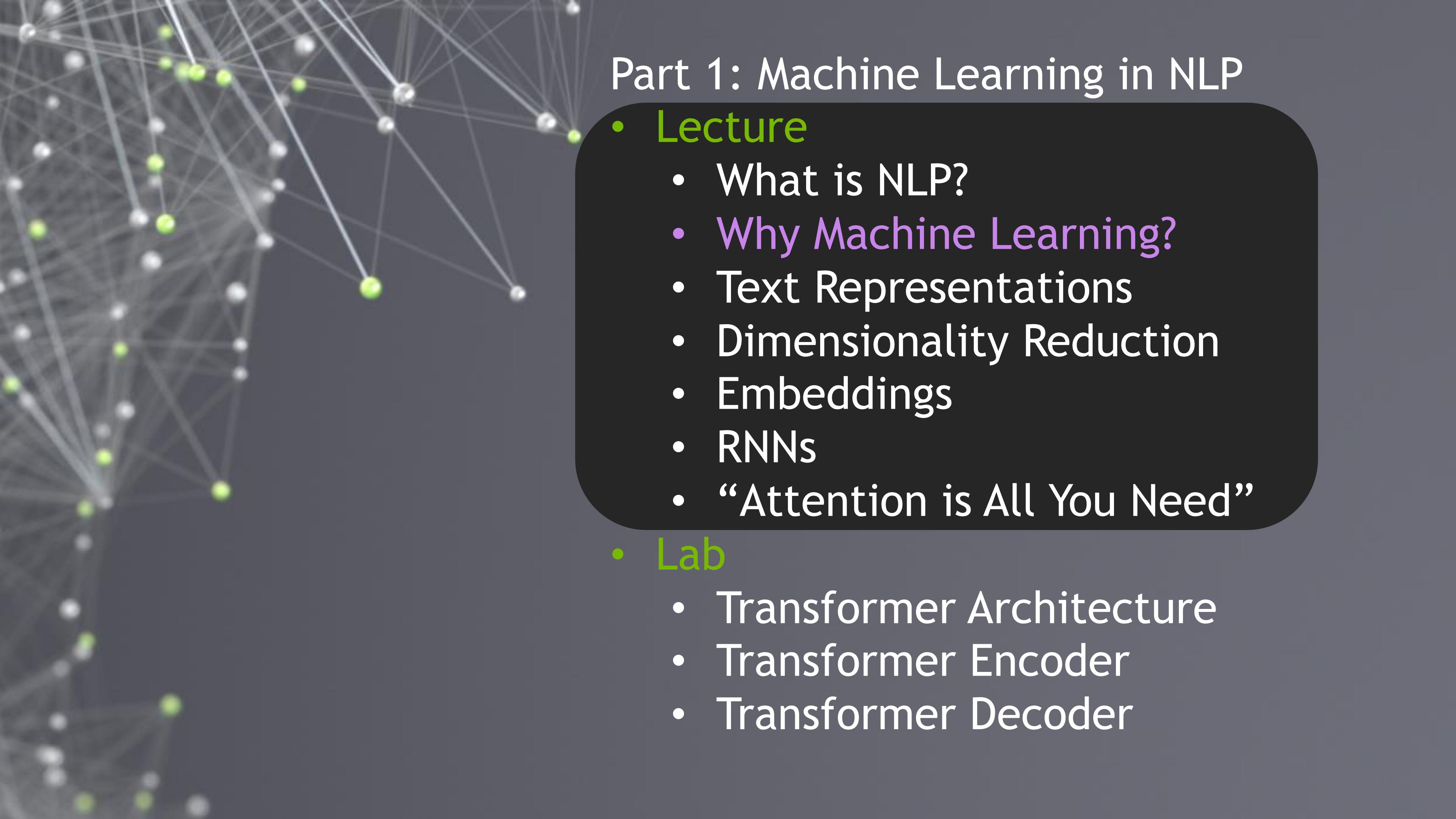


Text to  
Speech

# NLP TASKS

“Tasks” refer to specific textual language applications

Machine Translation	Sentiment Analysis	Question Answering	Automatic Text Summarization
Author Attribution	Named Entity Recognition	Text Classification	Spell Checking
And many more!			



# Part 1: Machine Learning in NLP

- **Lecture**
  - What is NLP?
  - Why Machine Learning?
  - Text Representations
  - Dimensionality Reduction
  - Embeddings
  - RNNs
  - “Attention is All You Need”
- **Lab**
  - Transformer Architecture
  - Transformer Encoder
  - Transformer Decoder

# MACHINE LEARNING FOR CV

What are these three images?



# THE CHALLENGE

Complexity of human language

- Languages often seem to behave in arbitrary ways and forms
- Ambiguity, sarcasm and irony are often not apparent from purely textual information
- Domain-specific terms and phrases that may not even be grammatically correct

# PRINCIPLES AND PARAMETERS

## Linguistic Concept

- ▶ Framework created by linguists Noam Chomsky and Howard Lasnik
- ▶ The framework states that languages are composed of ‘hard-wired’ principles and language-specific instantiations
- ▶ In software terms: A ‘language’ is an object, with different implementations of virtual functions. We compute a binary value - whether a sentence is grammatical or not - by running it through a language-specific Chain-of-Responsibility structure

# PRINCIPLES AND PARAMETERS

## Example: Word Order

- ▶ English: John ate apples
  - ▶ Japanese: Jon wa ringo o tabeta
- John                      apple                      ate

# PRINCIPLES AND PARAMETERS

Example: Null Subject

- ▶ English:      It      is      raining
- ▶ Spanish:      Está      lloviendo  
                      is      raining
- ▶ What is the role of ‘It’ in English?

# SYNTACTIC AMBIGUITY

Linguistic Concept

- ▶ John and Henry's parents arrived at the house.
- ▶ How many people arrived in total?

# THE CHALLENGE

When viewed through the eyes of a linguist

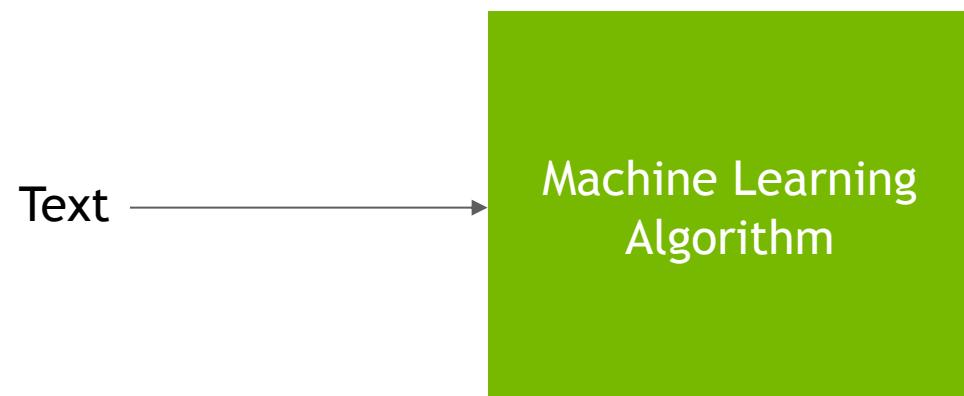
- Languages obeys strict rules and generalizations
- These generalizations map nicely to software constructs that are very helpful when we design NLP systems
- The huge mass of textual data available today means that Machine Learning is an ideal approach for NLP



# PROBLEM FORMULATION

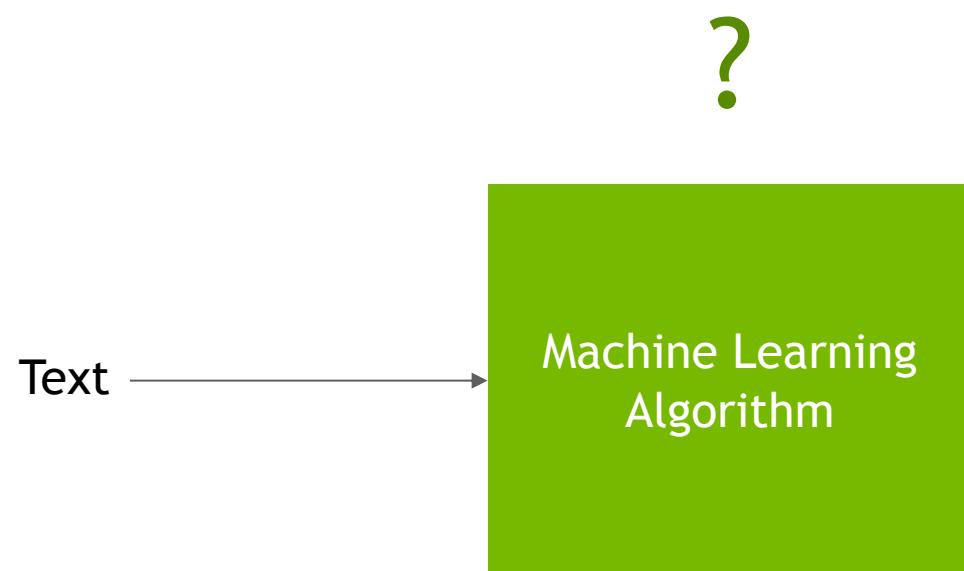
# MACHINE LEARNING

Discovering the discussed structures in text



# MACHINE LEARNING

Discovering the discussed structures in text



# MACHINE LEARNING

Design decisions

?

Problem formulation

?

?

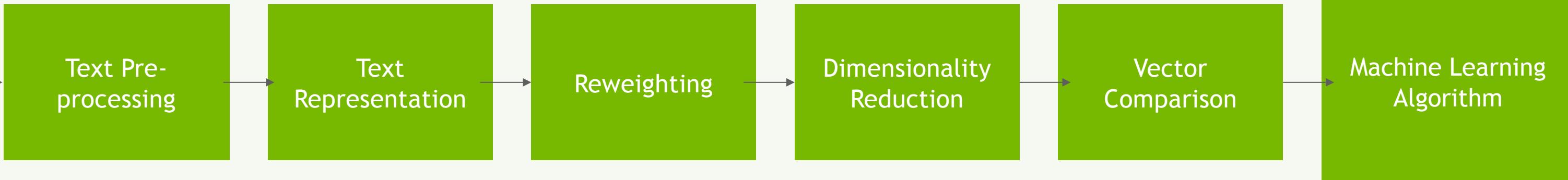
?

?

?

?

Text

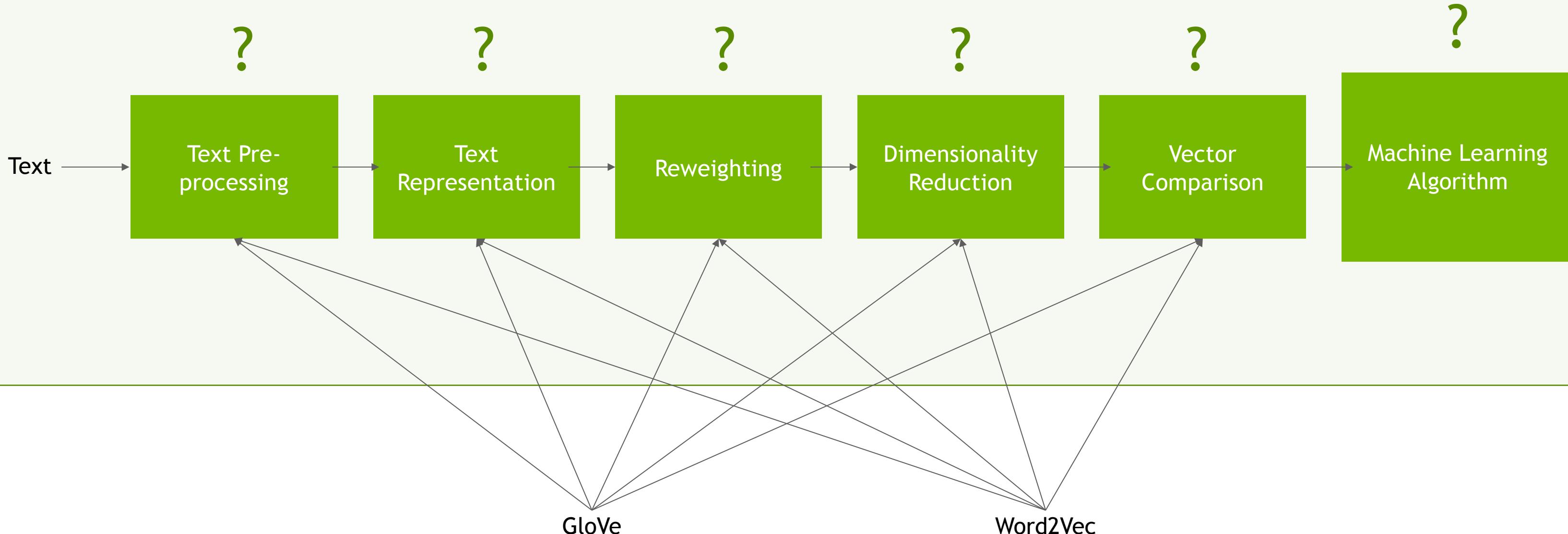


# MACHINE LEARNING

All linear combinations feasible

?

Problem formulation



# MACHINE LEARNING

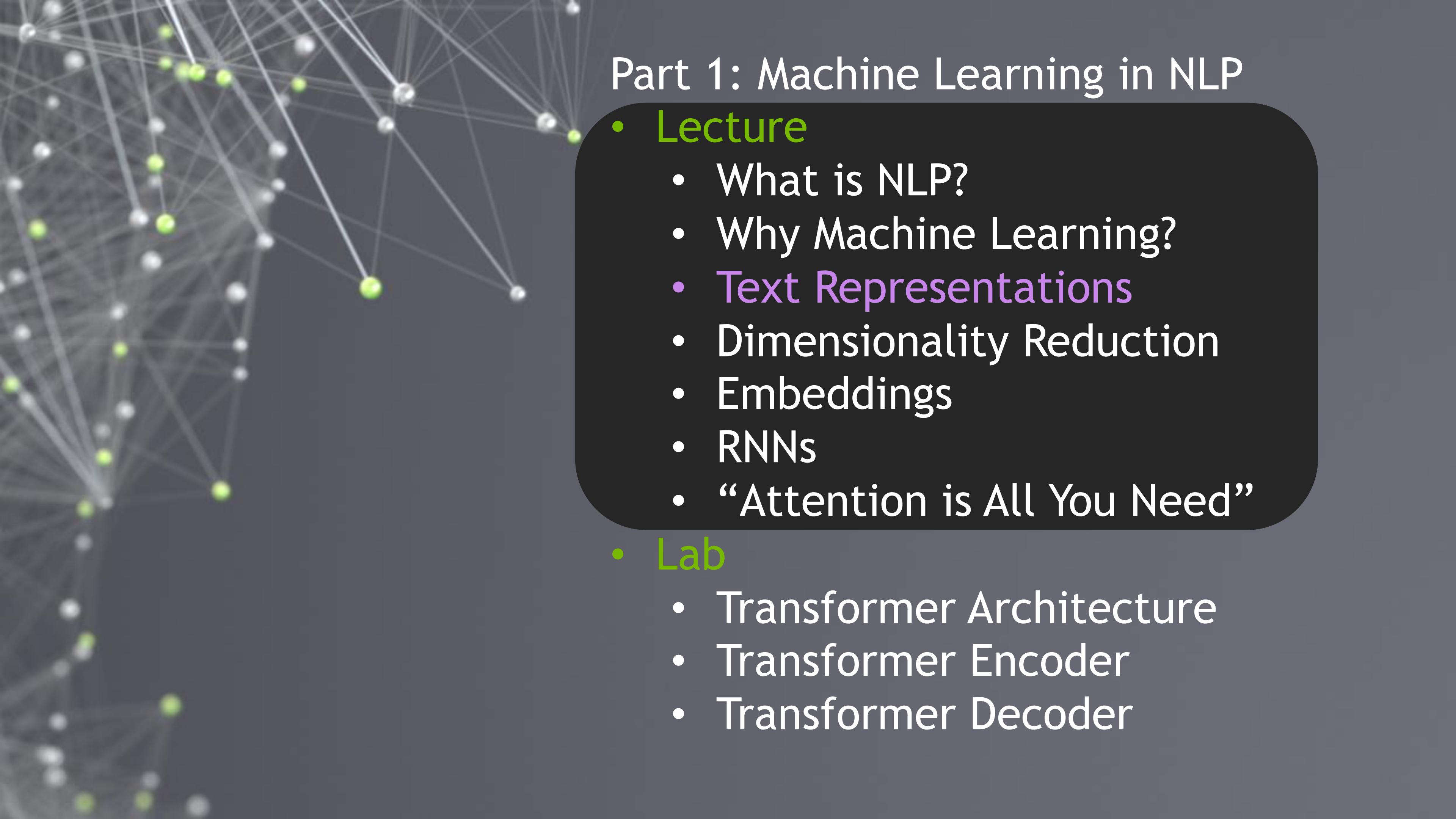
## In this class

Subset of Problem formulations



Subset of word representations

Subset of approaches



# Part 1: Machine Learning in NLP

- **Lecture**
  - What is NLP?
  - Why Machine Learning?
  - **Text Representations**
  - Dimensionality Reduction
  - Embeddings
  - RNNs
  - “Attention is All You Need”
- **Lab**
  - Transformer Architecture
  - Transformer Encoder
  - Transformer Decoder

# TEXT REPRESENTATIONS

## The bag of words

- Bag of words/ngrams - feature per word/ngram

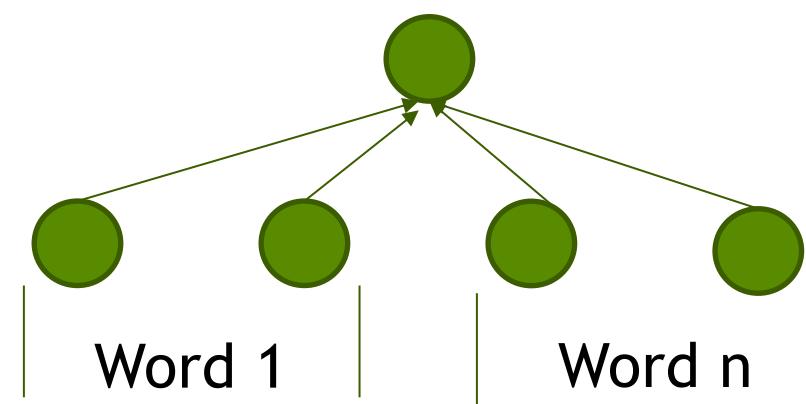
the cat sat on the mat

cat	sat	on	the	mat	quic kly	...  Vocabulary
1	1	1	2	1	0	

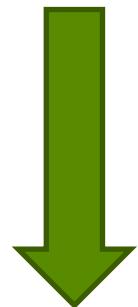
# THE BAG OF WORDS

## Key challenges

- ▶ Sparse Input (1-hot)



$p \gg n$  (overfitting!)



- ▶ No semantic generalization
- ▶ *dog*: 1 0 0 0 0 ... 0
- ▶ *cat*: 0 0 1 0 0 ... 0



lots of data required,  
low accuracy



DISTRIBUTED WORD  
REPRESENTATIONS

# DISTRIBUTIONAL HYPOTHESIS

## The intuition

*'You can tell a word by the company it keeps'*

Firth 1957

*'Distributional statements can cover all of the material of a language without requiring support from other types of information'*

Harris 1954

*'The meaning of a word is its use in the language'*

Wittgenstein 1953

*'The complete meaning of a word is always contextual, and no study of meaning apart from context can be taken seriously.'*

Firth 1957

# CO-OCCURRENCE PATTERNS

The latent information

	a	big	bug	the	little	but	beetle	bit	back
a	0	5	4	2	1	0	0	3	0
big	5	0	10	8	4	0	4	8	4
bug	4	10	0	8	4	0	4	8	5
the	2	8	8	0	8	3	8	10	3
little	1	4	4	13	1	3	10	8	0
but	0	0	0	7	7	0	7	3	0
beetle	0	4	4	11	11	4	1	8	1
bit	3	8	7	12	9	3	8	0	1
back	0	4	5	3	0	0	1	2	0

# CO-OCCURRENCE PATTERNS

The latent information

*The cat sat on the mat*

*The dog sat on the mat*

*The elephant sat on the mat*

*The quickly sat on the mat*

# CO-OCCURRENCE PATTERNS

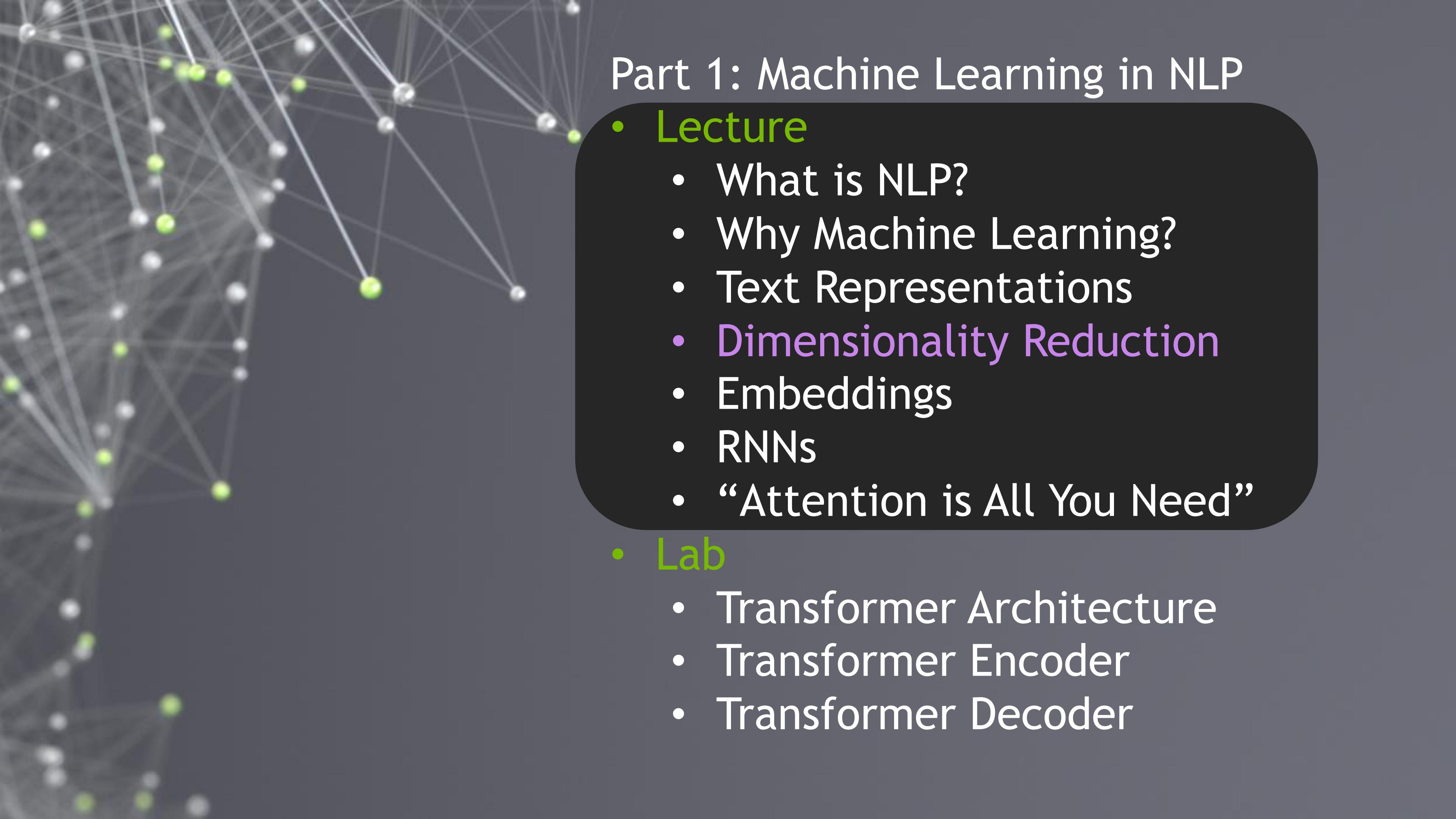
Where to find them?

Possible relationships:

- Word to documents (very sparse and very wide)
- Word to word (very dense and compact)
- Word to user / person
- Word to user behaviour
- Word to product
- Word to custom feature (e.g. movie raking)

Not only metrics:

- Word to user to product

A faint, semi-transparent background network graph consisting of numerous small white and green circular nodes connected by thin white lines, representing a complex system or data structure.

# Part 1: Machine Learning in NLP

- **Lecture**
  - What is NLP?
  - Why Machine Learning?
  - Text Representations
  - Dimensionality Reduction
  - Embeddings
  - RNNs
  - “Attention is All You Need”
- **Lab**
  - Transformer Architecture
  - Transformer Encoder
  - Transformer Decoder

# DIMENSIONALITY REDUCTION

## Rationale

The need for compact and computationally efficient representations

More robust notions of distance exposing the information captured by our distributional representation

A complex network graph is displayed against a dark gray background. The graph consists of numerous small, semi-transparent white and light green circular nodes, connected by a dense web of thin, gray lines representing edges. The nodes are distributed across the frame, with a higher density in the upper half and some isolated or smaller clusters in the lower half. The overall effect is one of a large, interconnected system.

LSA

# LSA

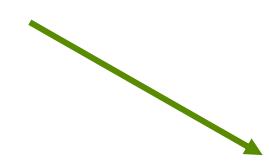
## Latent Semantic Analysis

?

# LSA

## Truncated SVD

Terms x Documents


$$X = T * S * P^T$$

# LSA

## Truncated SVD

Terms x Documents

$$X = T * S * P^T$$

K largest singular values

$$X = T_k * S_k * P_k^T$$

# LSA

## Truncated SVD

Terms x Documents

$$X = T * S * P^T$$

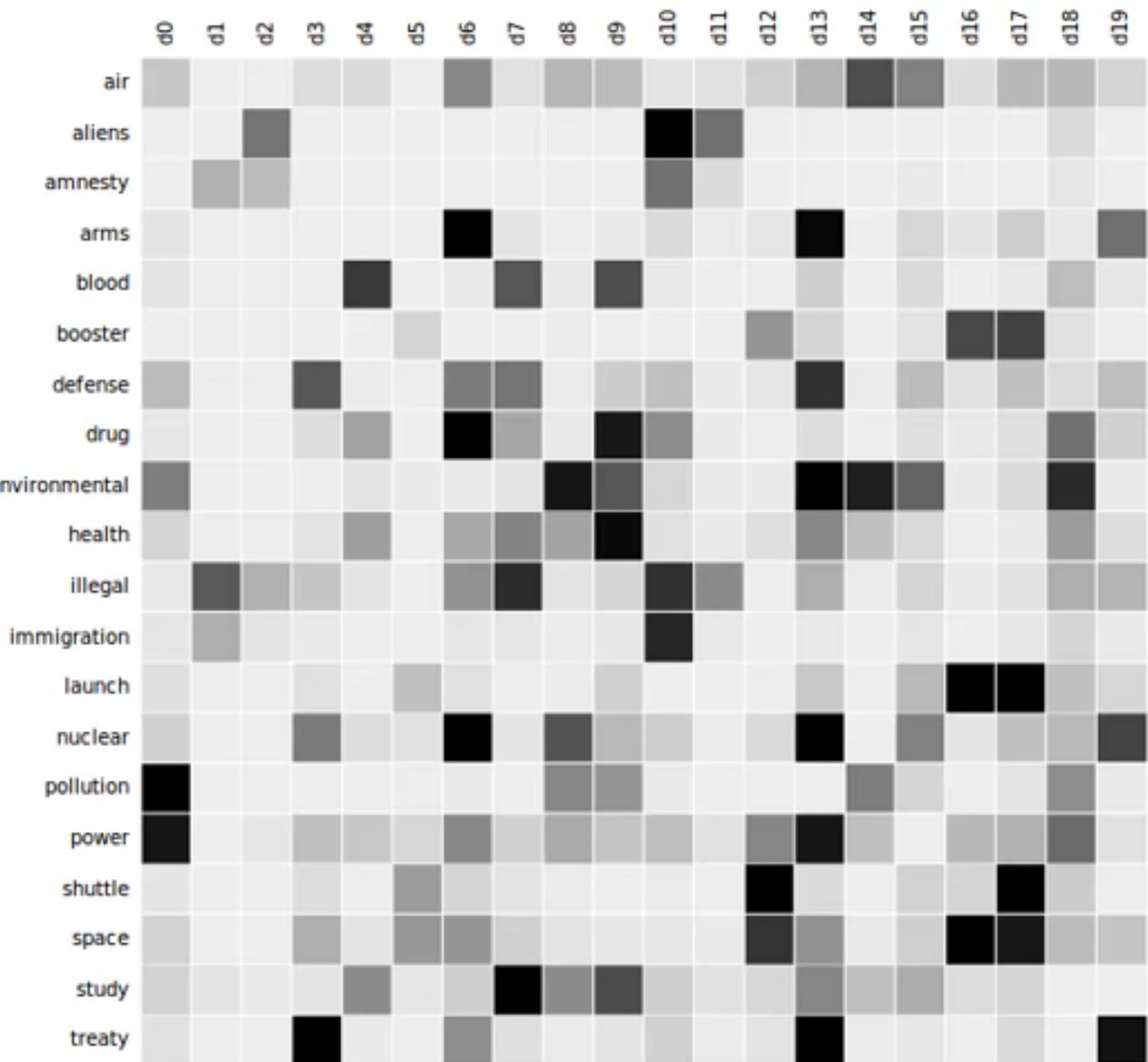
K largest singular values

$$X = T_k * S_k * P_k^T$$

Latent Semantic Space

# LSA

## Example



# LSA

## Question

What about large matrices?

What about complex corpora?

# PROBABILISTIC LSA

Statistical model which has been called aspect model

$$P(d, w) = \sum_{z \in \mathcal{Z}} P(z) P(d|z) P(w|z)$$

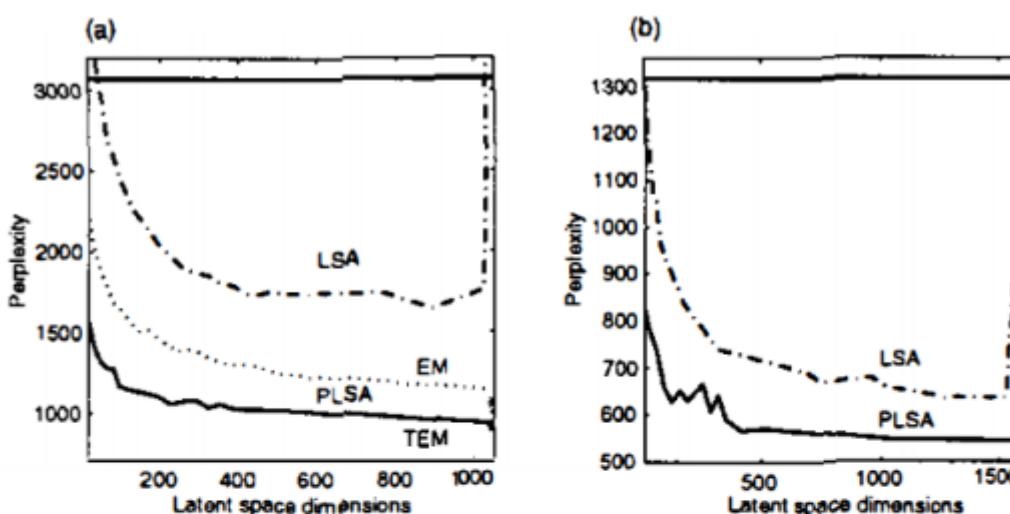
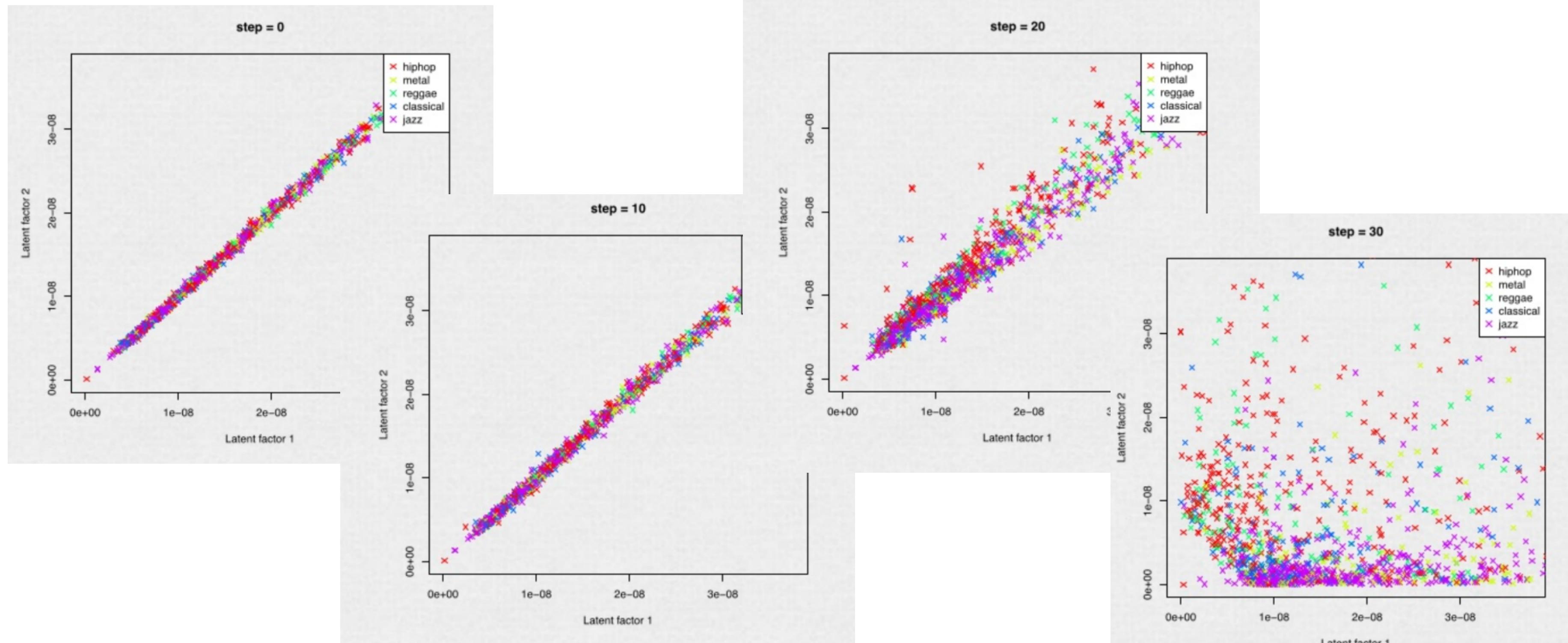


Figure 5: Perplexity results as a function of the latent space dimensionality for (a) the MED data (rank 1033) and (b) the LOB data (rank 1674). Plotted results are for LSA (dashed-dotted curve) and PLSA (trained by TEM = solid curve, trained by early stopping EM = dotted curve). The upper baseline is the unigram model corresponding to marginal independence. The star at the right end of the PLSA denotes the perplexity of the largest trained aspect models ( $K = 2048$ ).

# PROBABILISTIC LSA

Very broadly used (Spotify Example)





LDA

# LDA

## Latent Dirichlet Allocation

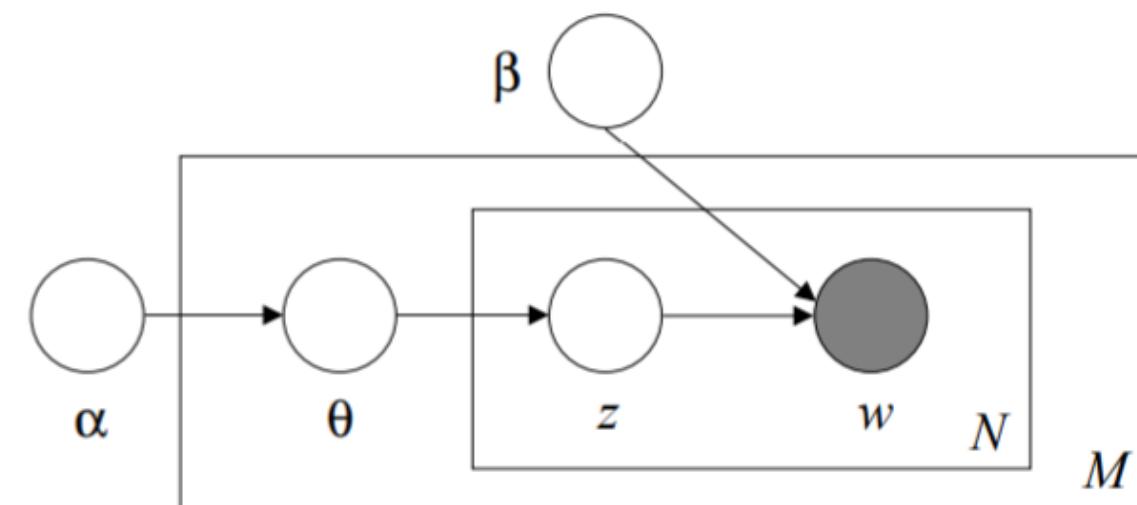
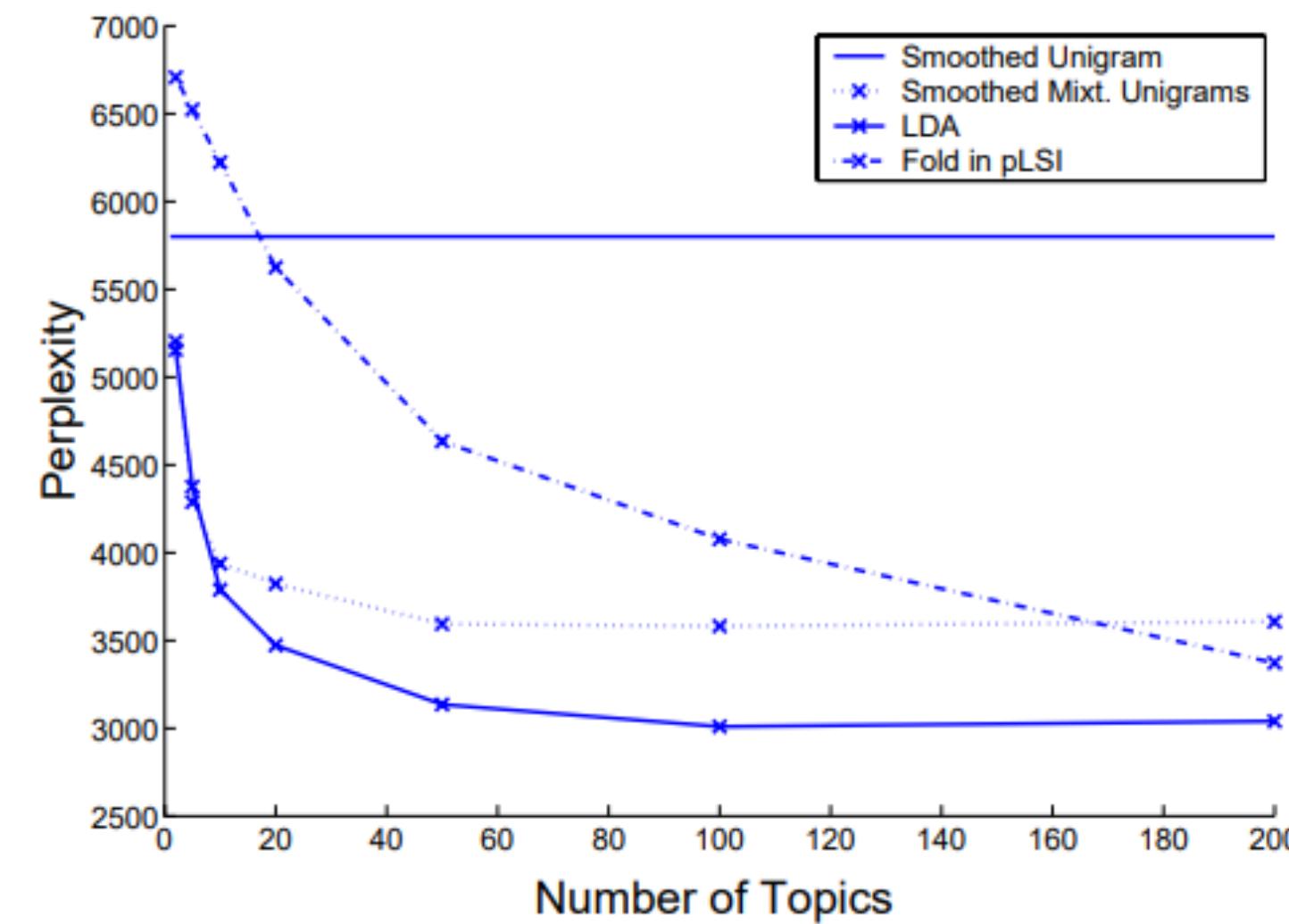
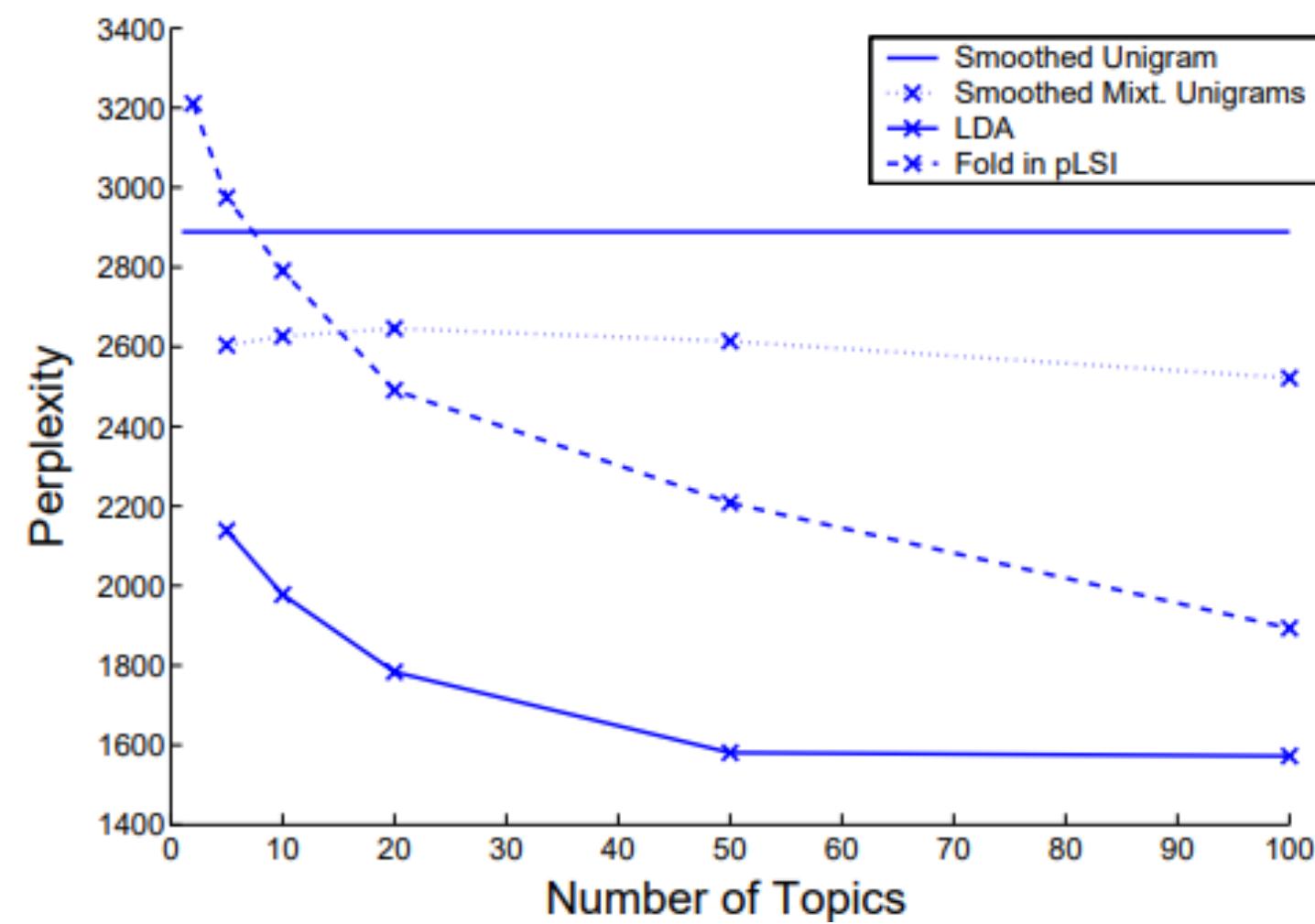


Figure 1: Graphical model representation of LDA. The boxes are “plates” representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

# LDA

## Latent Dirichlet Allocation

Perplexity results on the nematode (Left) and AP (Right) corpora for LDA, the unigram model, mixture of unigrams, and pLSI.





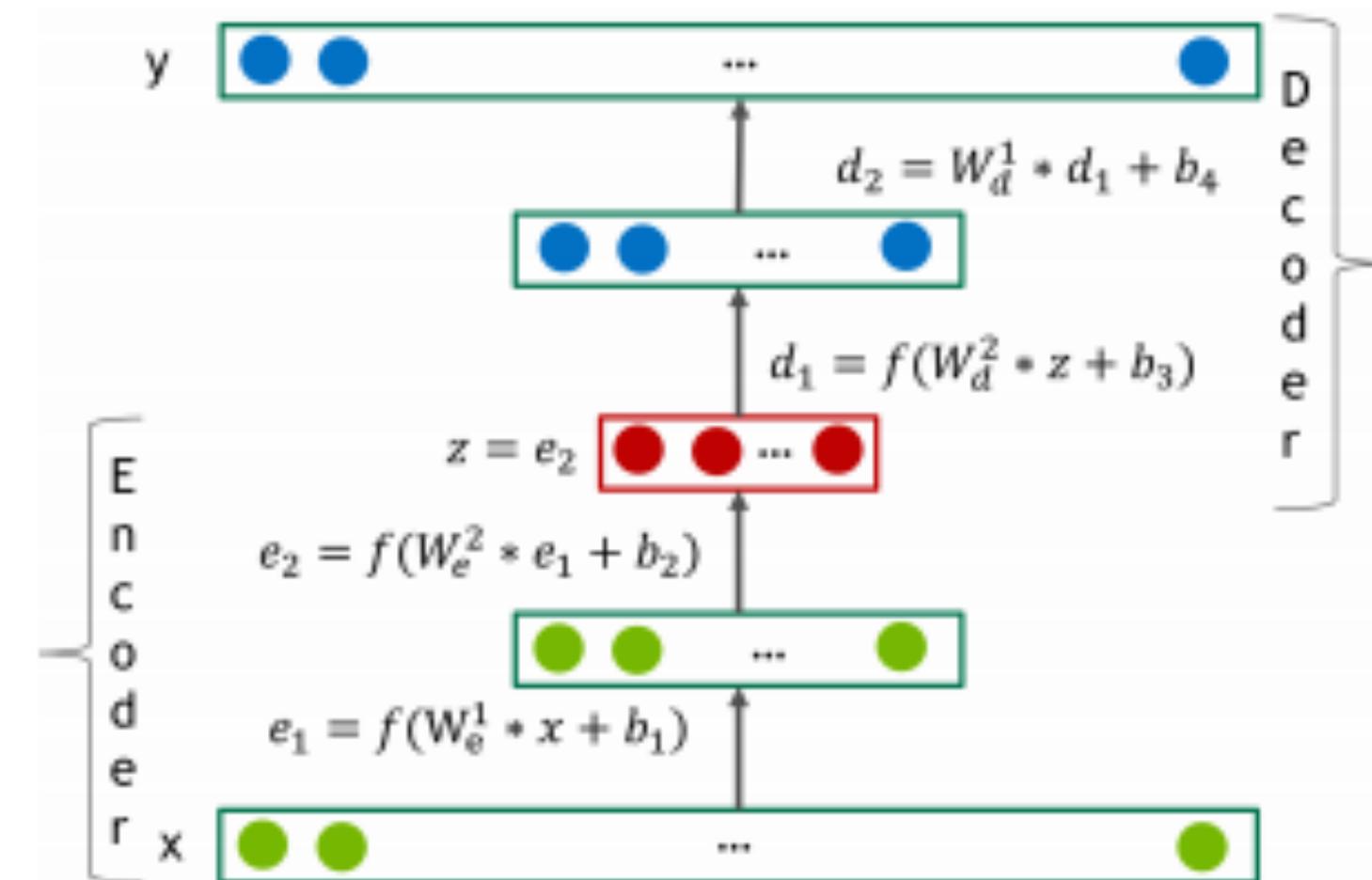
# AUTOENCODERS FOR DIMENSIONALITY REDUCTION

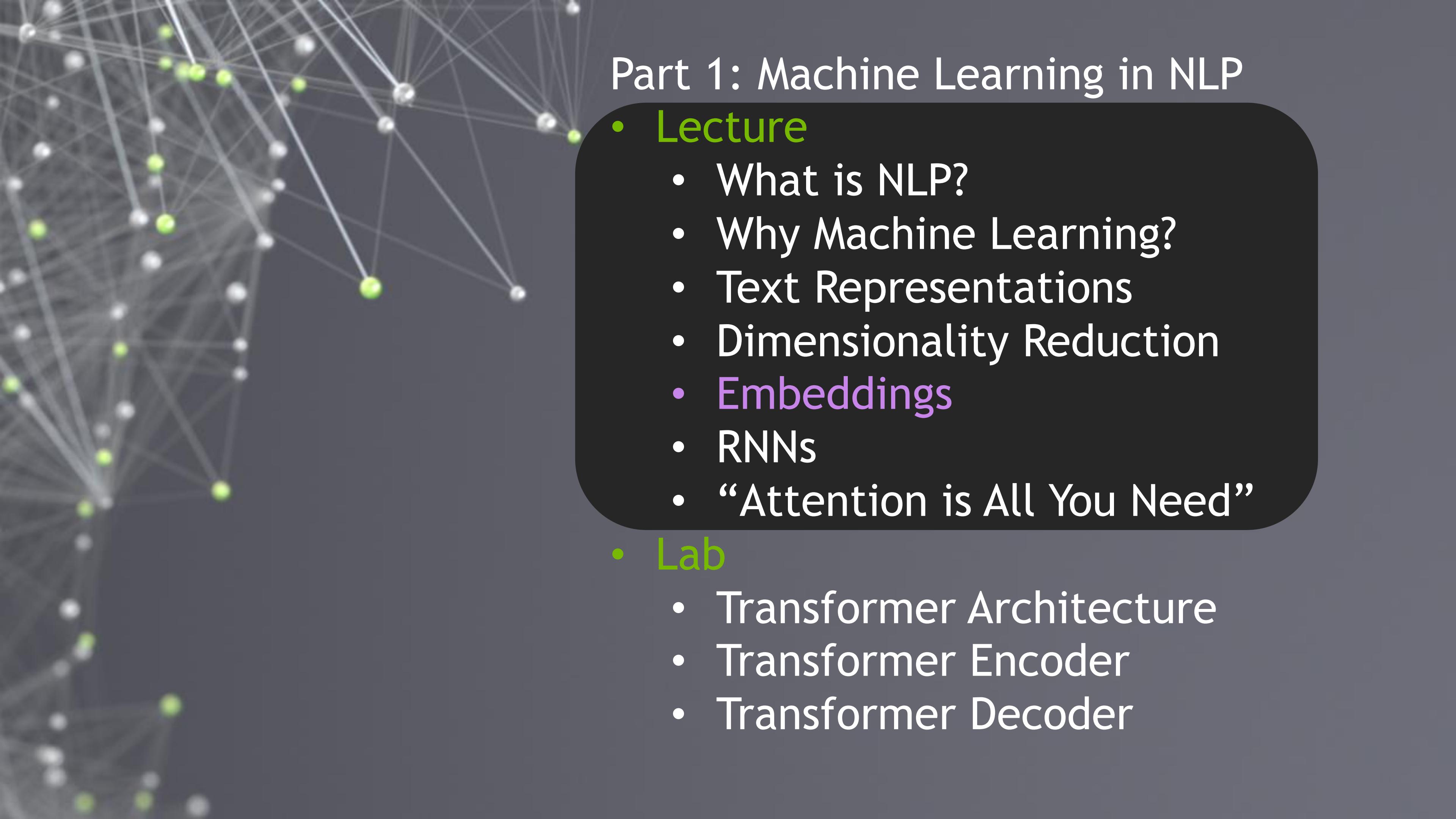
# AUTOENCODERS

Where to find them?

“An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner.”

Kramer, 1991





# Part 1: Machine Learning in NLP

- **Lecture**
  - What is NLP?
  - Why Machine Learning?
  - Text Representations
  - Dimensionality Reduction
  - **Embeddings**
  - RNNs
  - “Attention is All You Need”
- **Lab**
  - Transformer Architecture
  - Transformer Encoder
  - Transformer Decoder

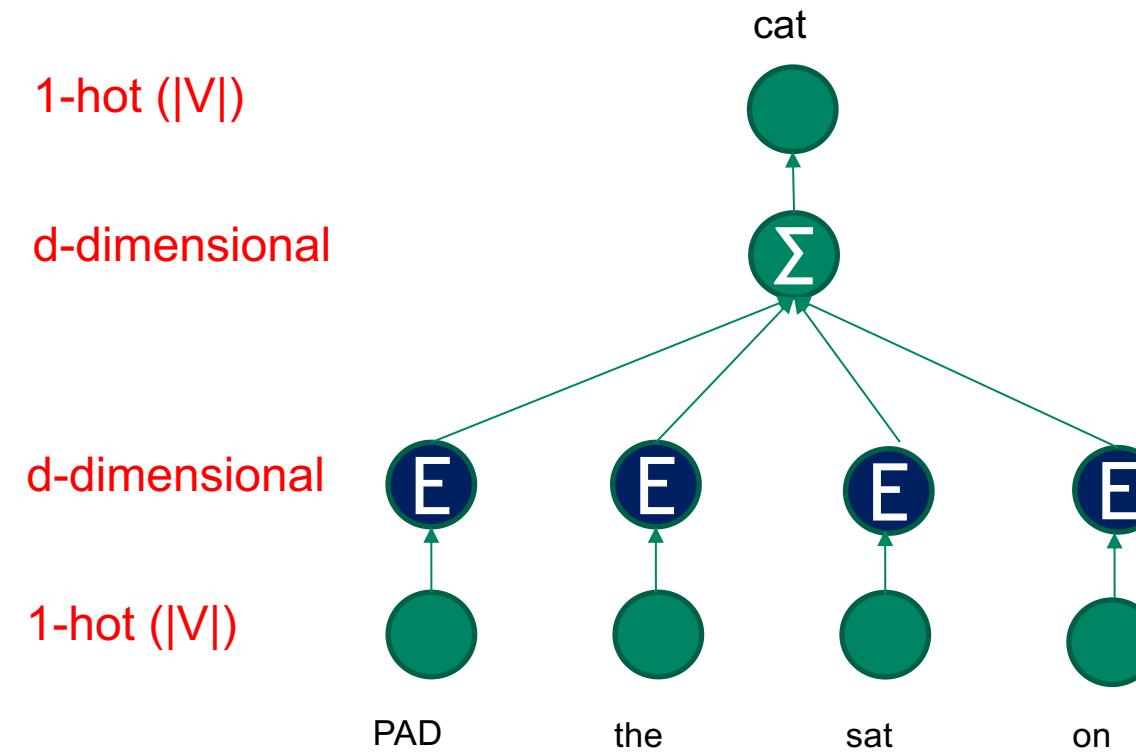
A complex network graph visualization on a dark background. The graph consists of numerous small, semi-transparent white and light green circular nodes connected by thin, semi-transparent grey lines. The nodes are distributed across the frame, with a higher density in the upper half and some isolated or smaller clusters in the lower half. The overall effect is a sense of a large, interconnected system.

WORD2VEC

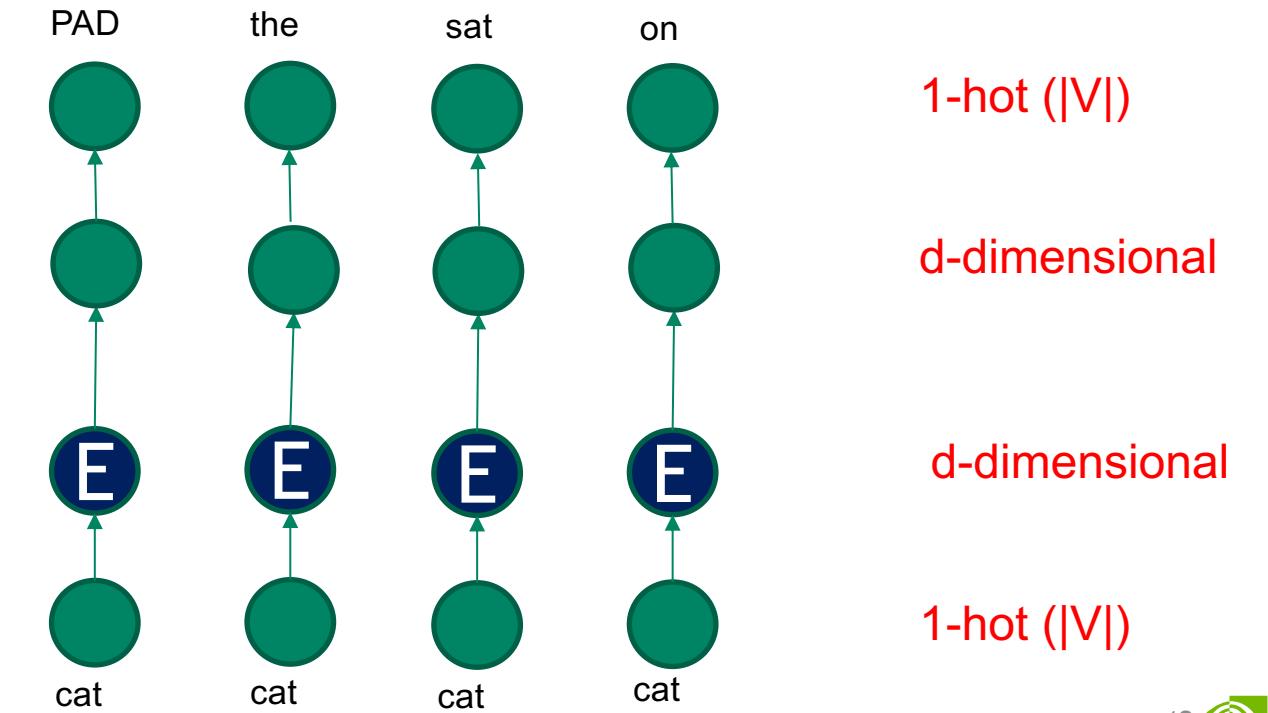
# WORD2VEC

- ▶ Mikolov et al., 2013 (while at Google)
- ▶ Linear model (trains quickly)
- ▶ Two models for training embeddings in an *unsupervised* manner:

Continuous Bag-of-Words (CBOW)



Skip-Gram



A network graph visualization on a dark gray background. The graph consists of numerous small, semi-transparent white circular nodes connected by thin, light gray lines representing edges. Several nodes are highlighted with a bright lime green color, creating a visual emphasis on specific data points or clusters. The overall structure appears organic and interconnected.

GLOVE

# GLOVE

## The objective

To learn vectors for words such that their dot product is proportional to their probability of co-occurrence

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k ice)$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k steam)$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$P(k ice)/P(k steam)$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

# GLOVE

## The objective

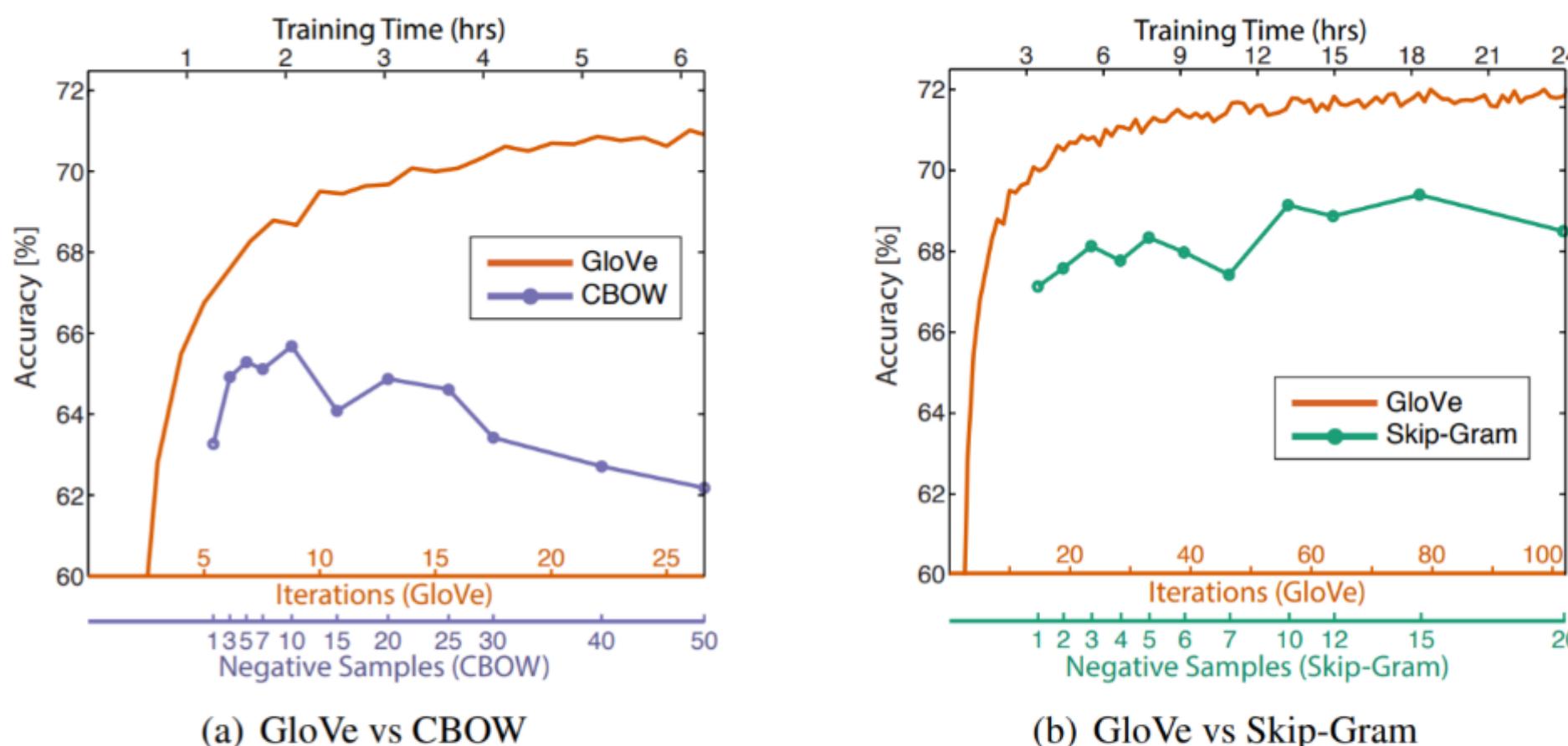
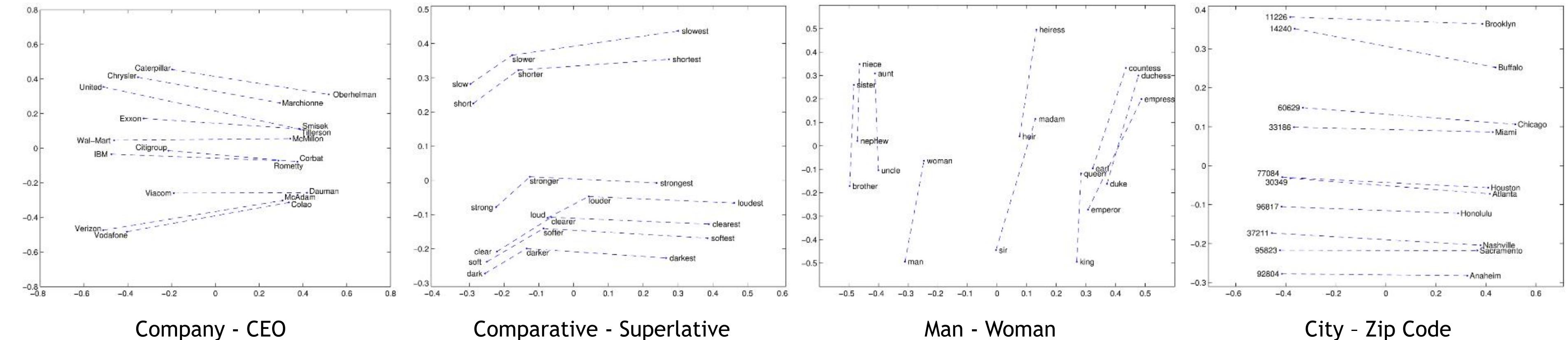


Figure 4: Overall accuracy on the word analogy task as a function of training time, which is governed by the number of iterations for GloVe and by the number of negative samples for CBOW (a) and skip-gram (b). In all cases, we train 300-dimensional vectors on the same 6B token corpus (Wikipedia 2014 + Gigaword 5) with the same 400,000 word vocabulary, and use a symmetric context window of size 10.

# GLOVE

## Properties



Company - CEO

Comparative - Superlative

Man - Woman

City - Zip Code



USING THE EMBEDDINGS

# MACHINE LEARNING

## In this class

Subset of Problem formulations



Subset of word representations

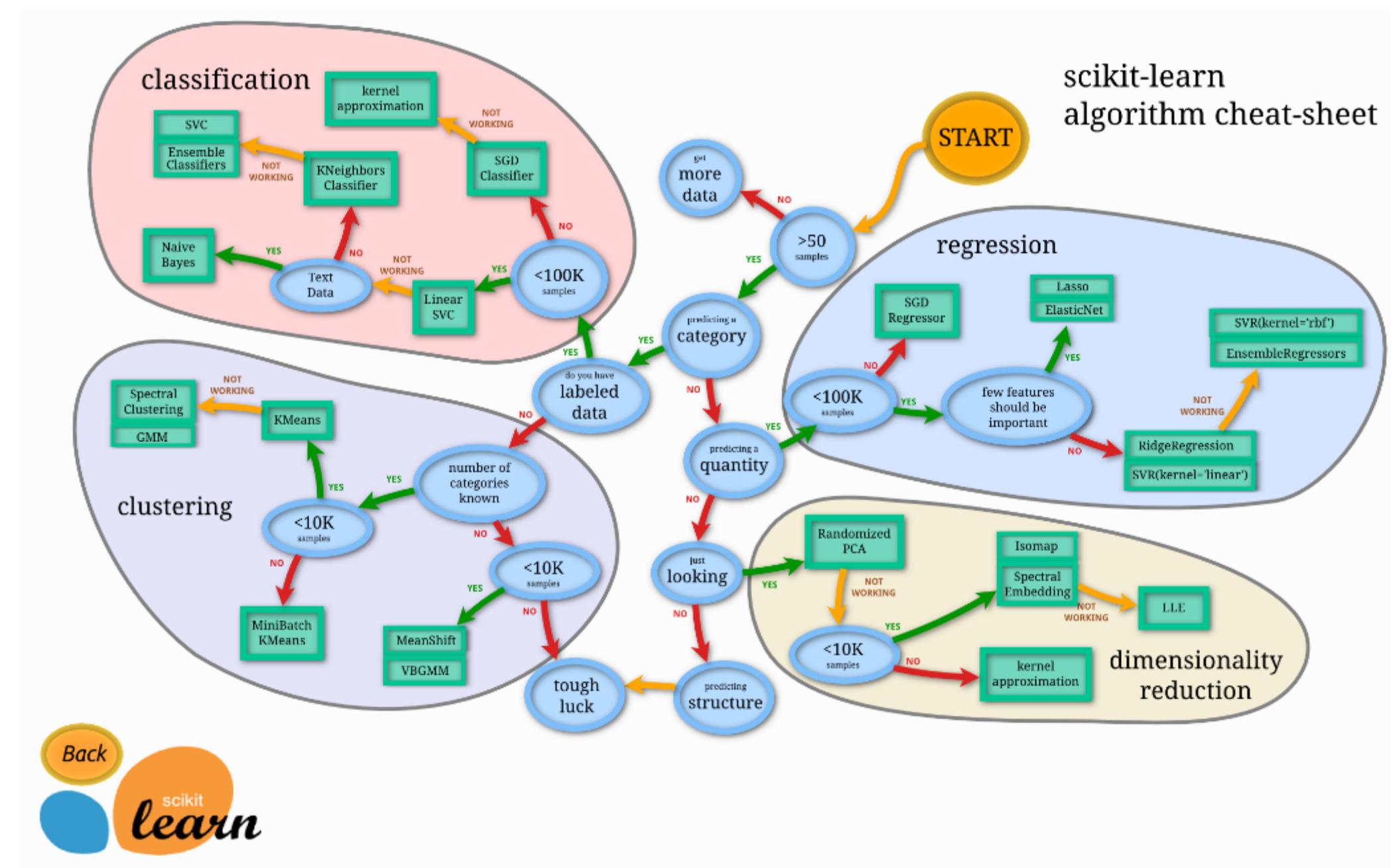
Subset of approaches



CLASSICAL APPROACHES

# CLASSICAL APPROACHES

## Very broad selection of tools





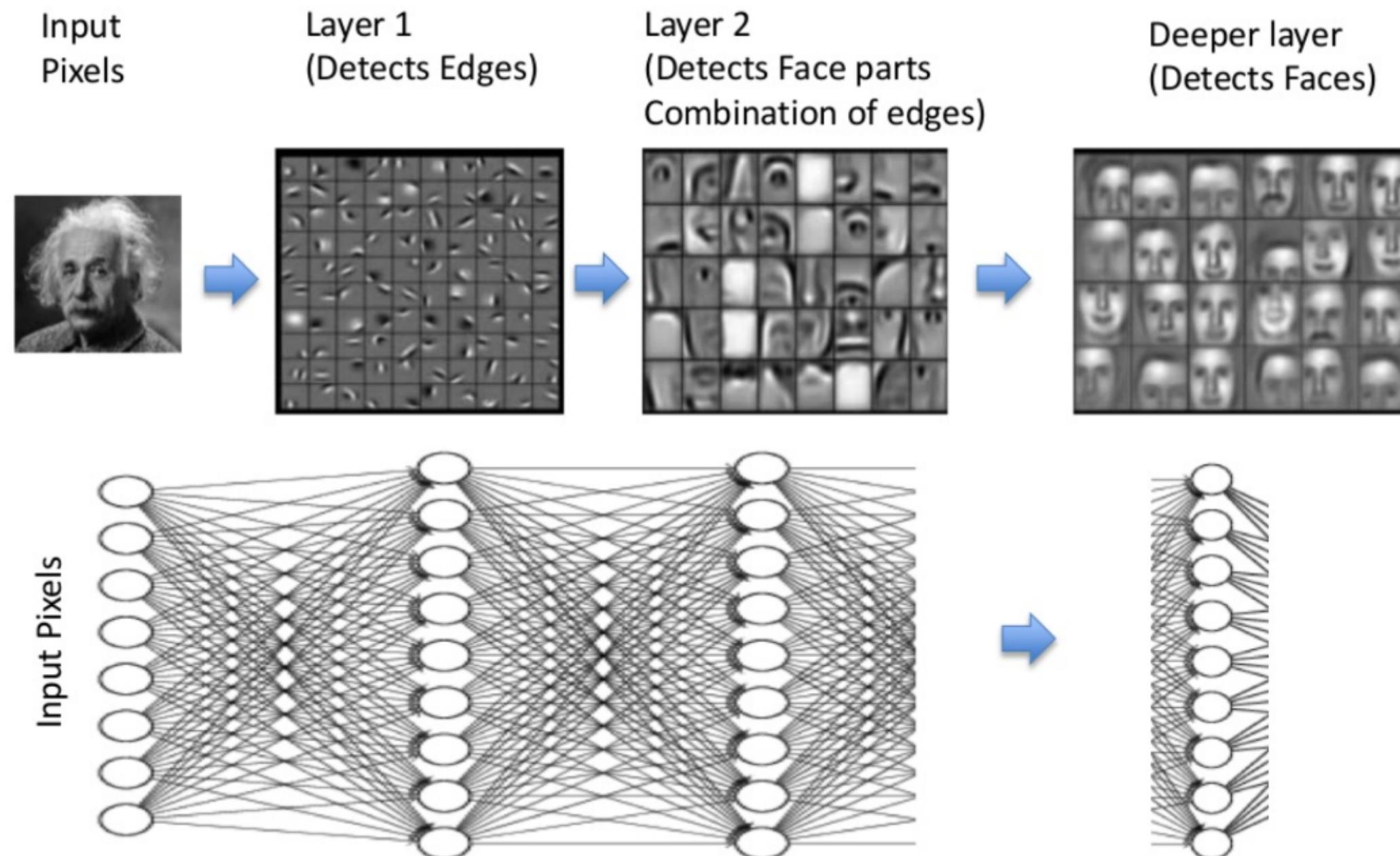
WHAT ABOUT FEATURE  
ENGINEERING?

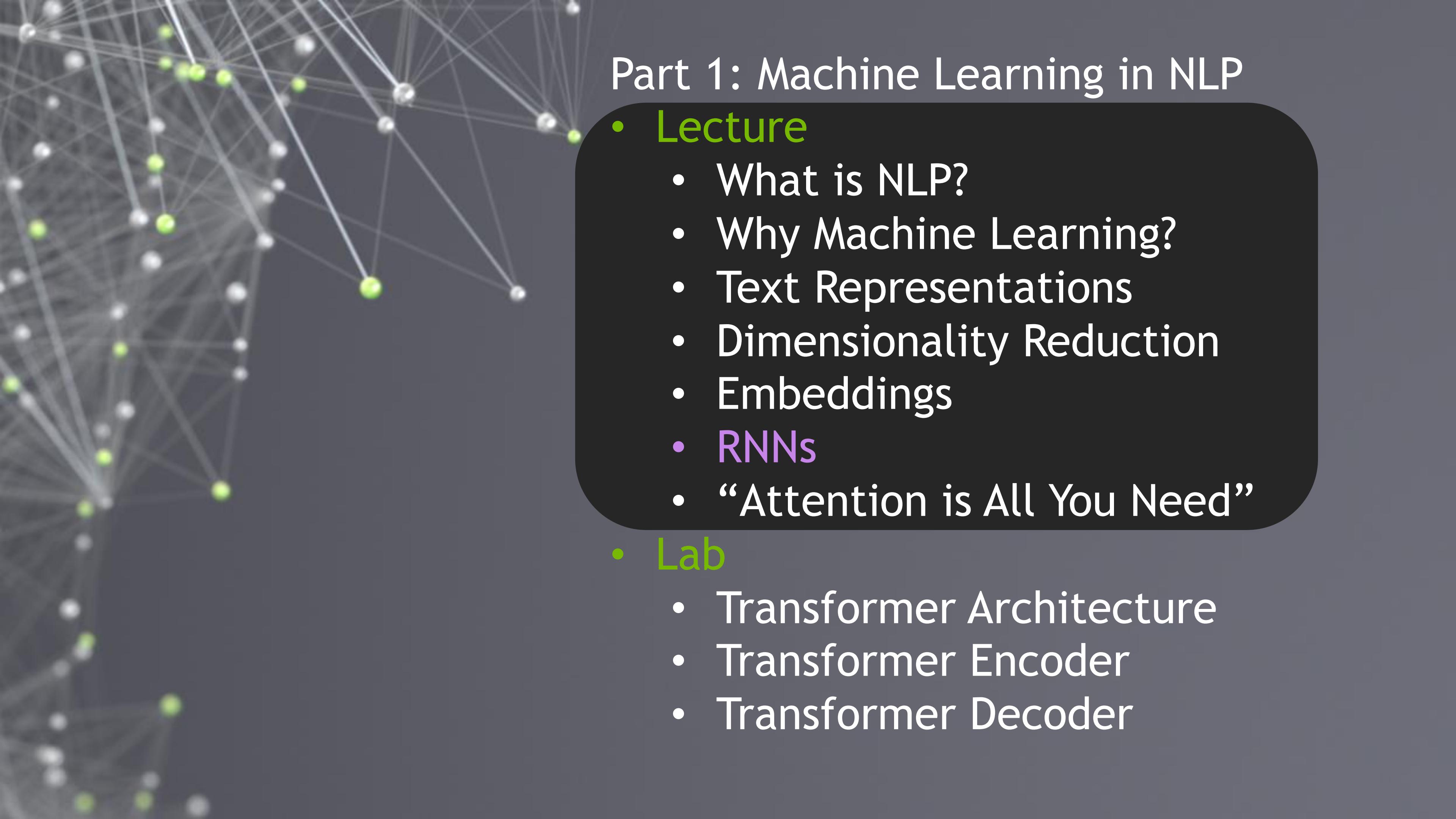


# DEEP REPRESENTATION LEARNING

# DEEP REPRESENTATION LEARNING

## Beyond distributional hypothesis



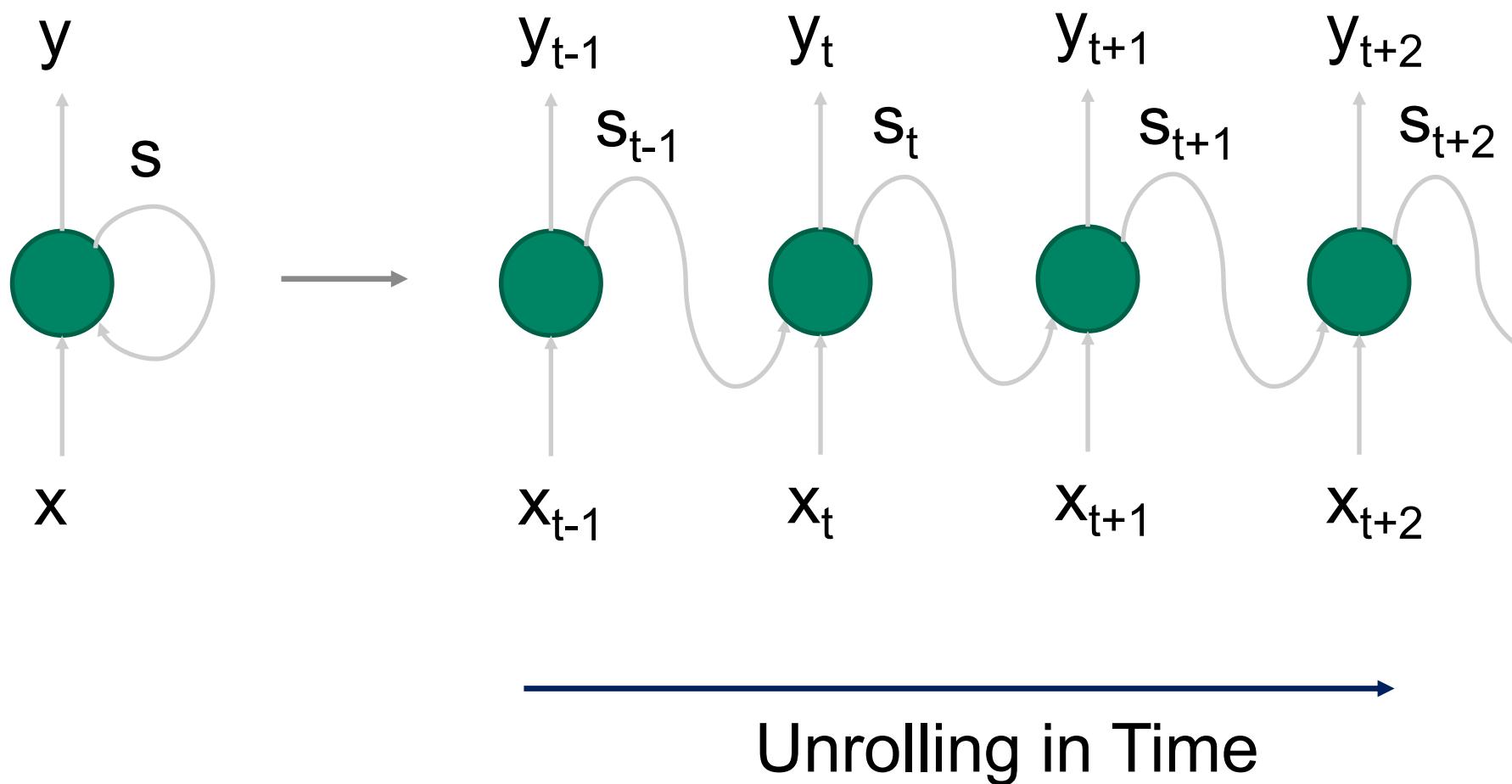


# Part 1: Machine Learning in NLP

- **Lecture**
  - What is NLP?
  - Why Machine Learning?
  - Text Representations
  - Dimensionality Reduction
  - Embeddings
  - RNNs
  - “Attention is All You Need”
- **Lab**
  - Transformer Architecture
  - Transformer Encoder
  - Transformer Decoder

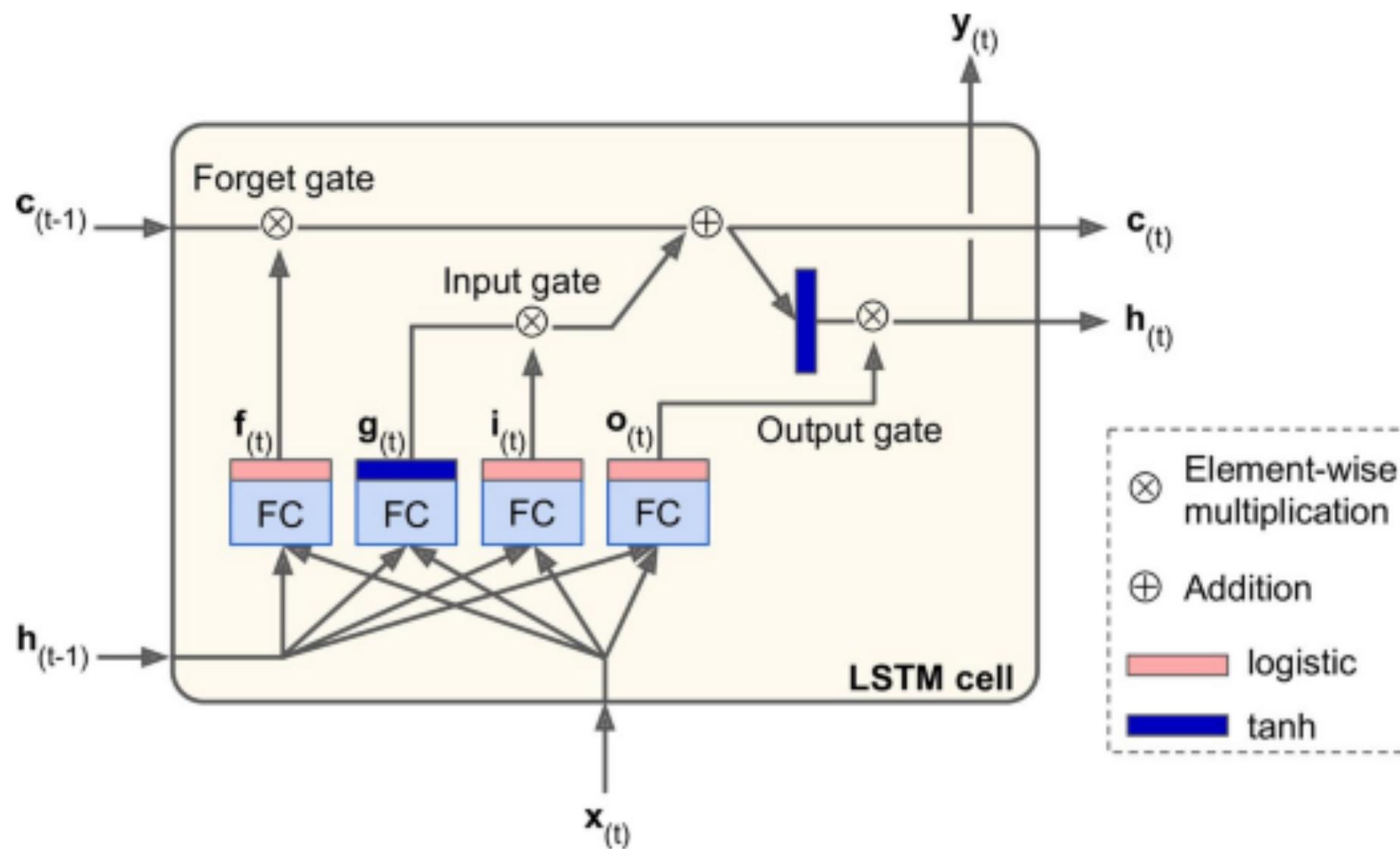
# RECURRENT NEURAL NETWORKS

## Basic principles



# LONG SHORT TERM (LSTM) CELL

Addressing problems of stability



$$\begin{aligned}\mathbf{i}_{(t)} &= \sigma(\mathbf{W}_{xi}^T \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hi}^T \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_i) \\ \mathbf{f}_{(t)} &= \sigma(\mathbf{W}_{xf}^T \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hf}^T \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_f) \\ \mathbf{o}_{(t)} &= \sigma(\mathbf{W}_{xo}^T \cdot \mathbf{x}_{(t)} + \mathbf{W}_{ho}^T \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_o) \\ \mathbf{g}_{(t)} &= \tanh(\mathbf{W}_{xg}^T \cdot \mathbf{x}_{(t)} + \mathbf{W}_{hg}^T \cdot \mathbf{h}_{(t-1)} + \mathbf{b}_g) \\ \mathbf{c}_{(t)} &= \mathbf{f}_{(t)} \otimes \mathbf{c}_{(t-1)} + \mathbf{i}_{(t)} \otimes \mathbf{g}_{(t)} \\ \mathbf{y}_{(t)} &= \mathbf{h}_{(t)} = \mathbf{o}_{(t)} \otimes \tanh(\mathbf{c}_{(t)})\end{aligned}$$

Legend:

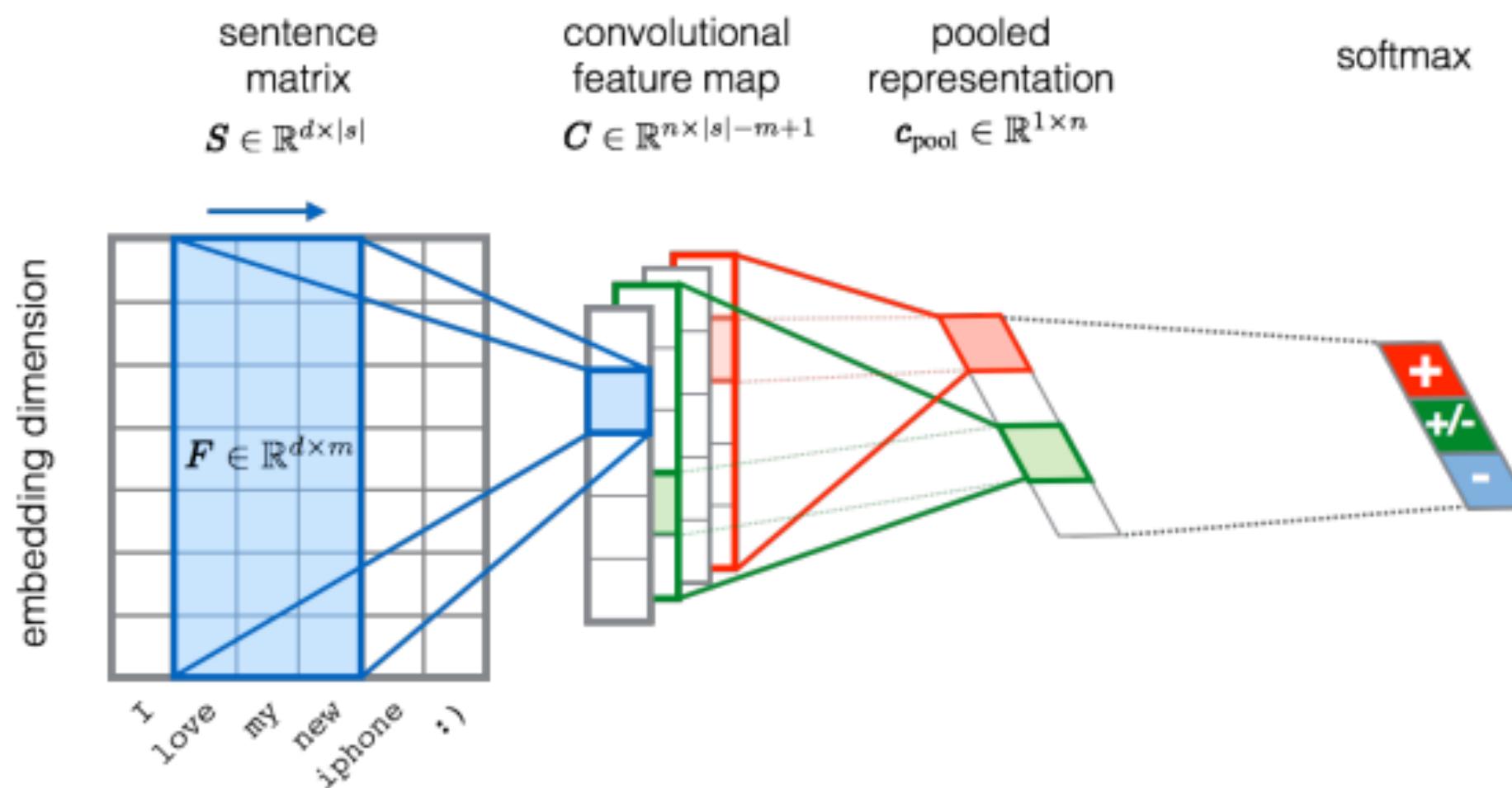
- $\otimes$  Element-wise multiplication
- $\oplus$  Addition
- Red box: logistic
- Blue bar: tanh

A complex network graph is displayed against a dark gray background. The graph consists of numerous small, semi-transparent white and lime green circular nodes connected by thin, light gray lines representing edges. The nodes are distributed across the frame, with a higher density in the upper half and some isolated or smaller clusters in the lower half. The overall effect is a sense of a large, interconnected system.

CNNs

# CONVOLUTIONAL NEURAL NETWORKS

## Basic principles



A complex network graph is displayed against a dark gray background. The graph consists of numerous small white dots representing nodes, connected by thin, semi-transparent gray lines representing edges. Interspersed among these are several larger, solid green dots, which appear to be highlighting specific nodes or clusters of nodes. The overall effect is one of a dense, interconnected system.

ATTENTION

# WHAT ABOUT LONG SEQUENCES?

The challenge illustrated with SQuAD

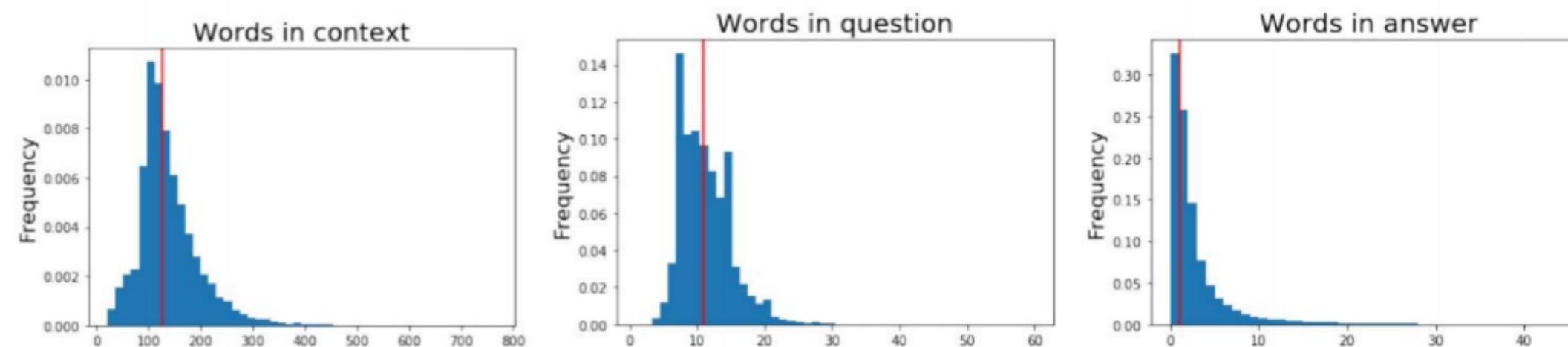


Figure 1: Number of words in contexts, questions, and answers in SQuAD training set.

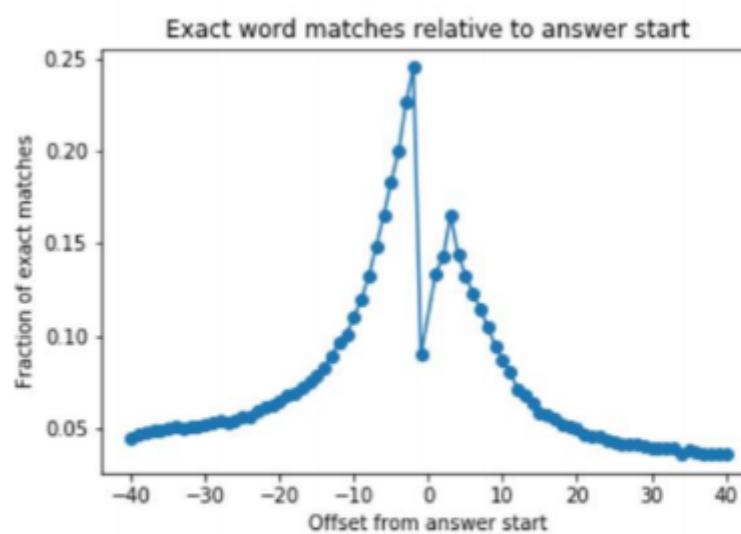
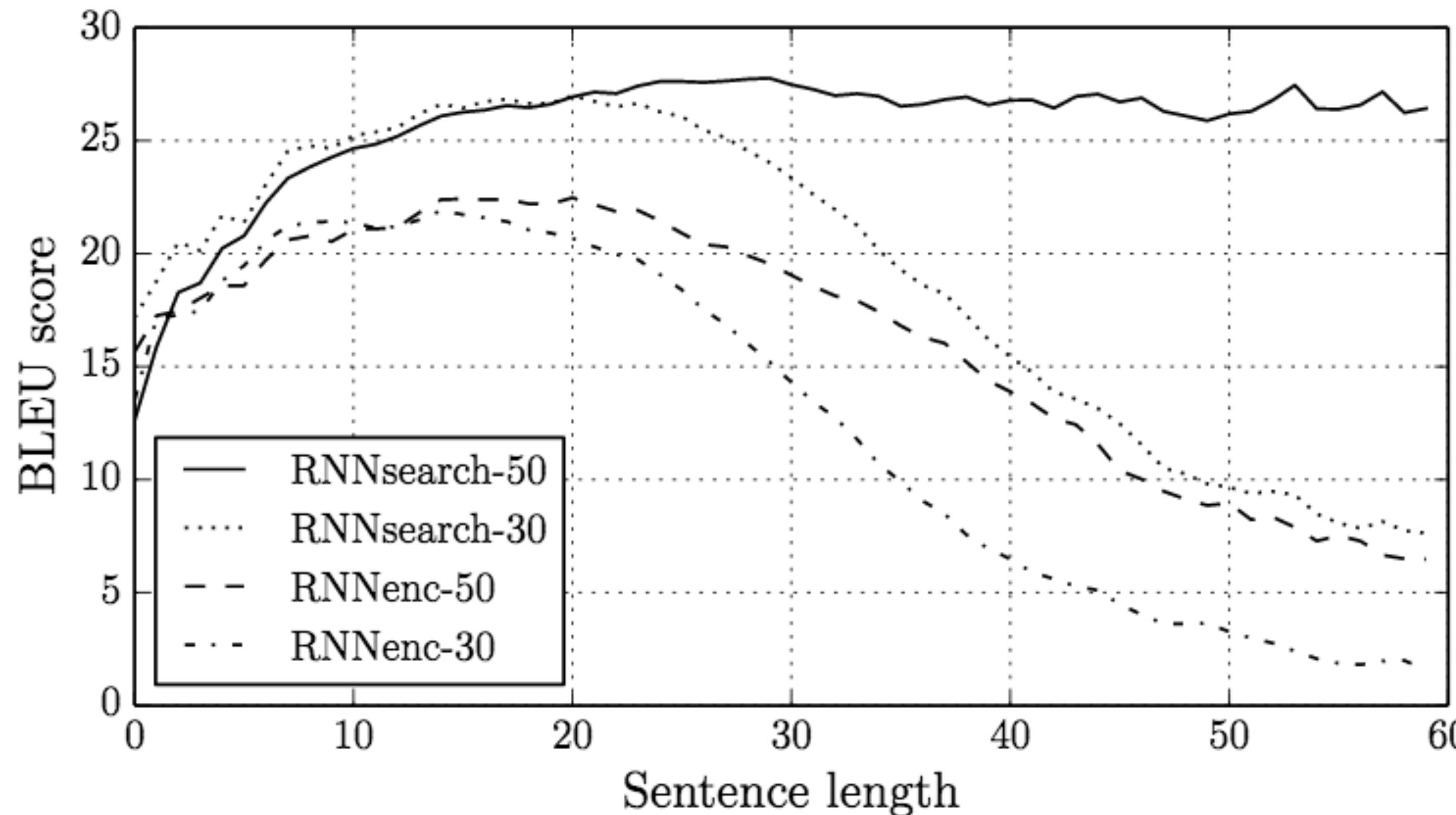


Figure 2: Frequency of exact word matches relative to answer start position

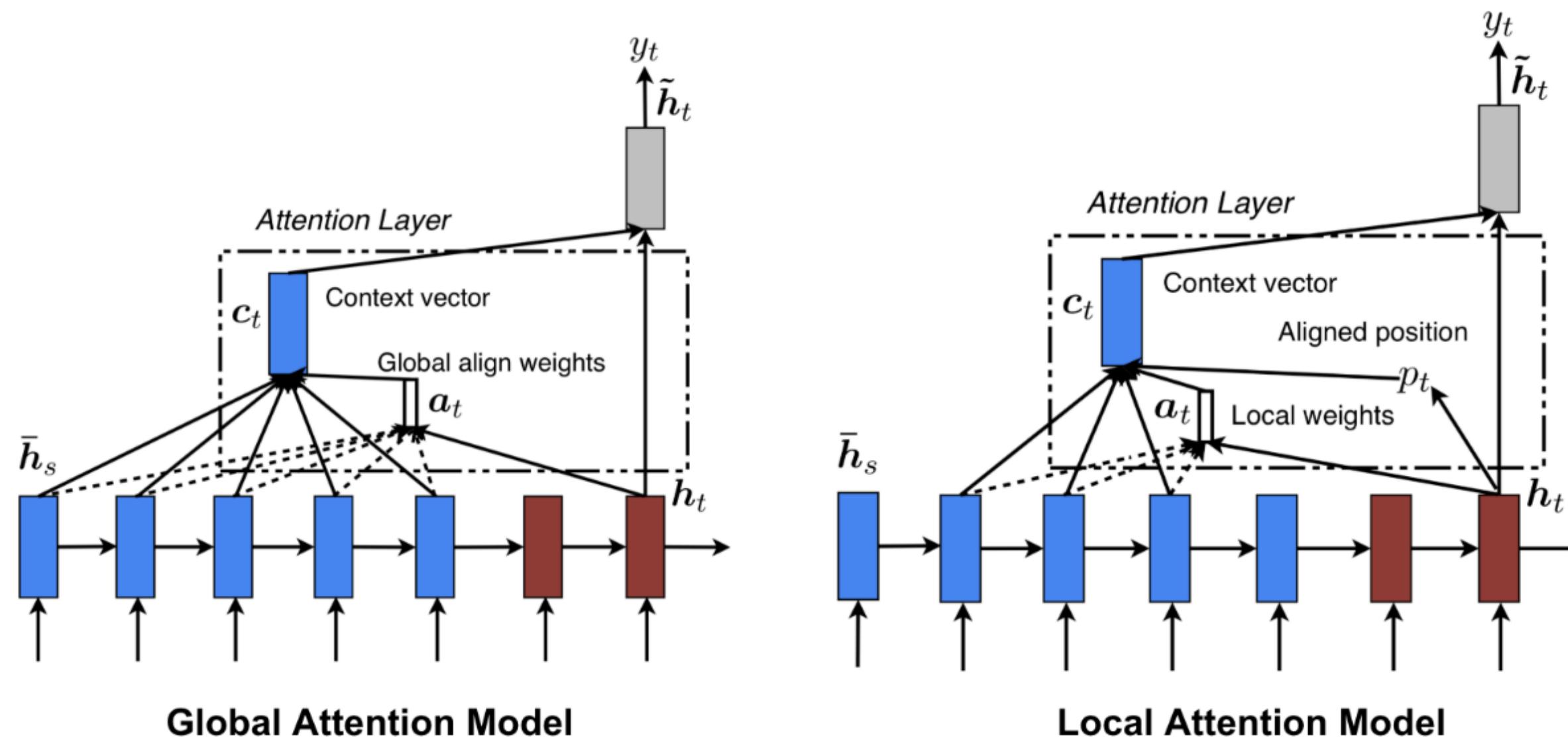
# WHAT ABOUT LONG SEQUENCES?

The challenge



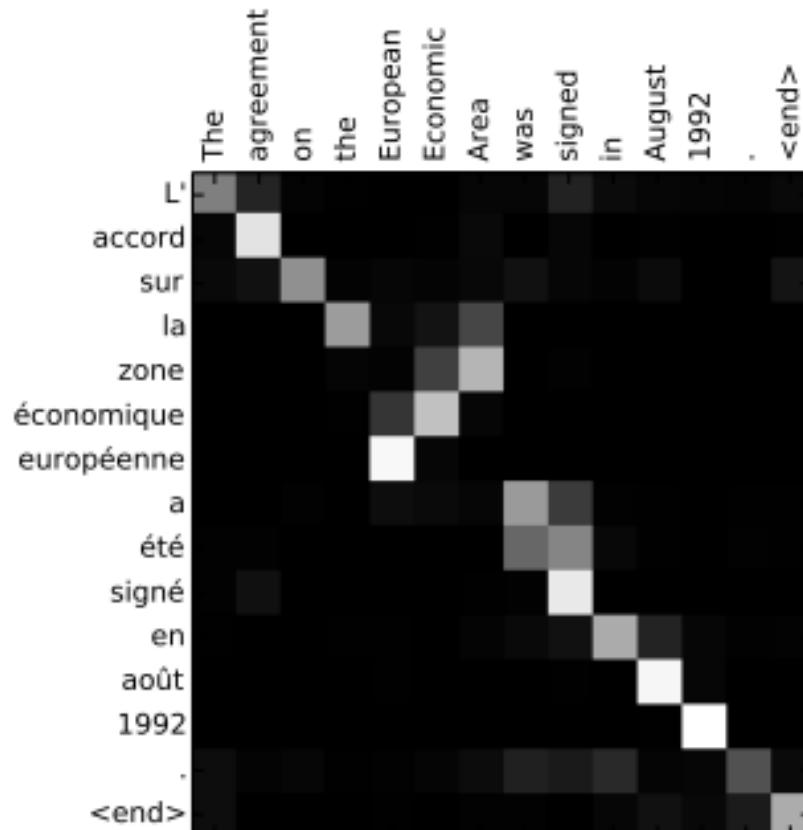
# ATTENTION

## The mechanism

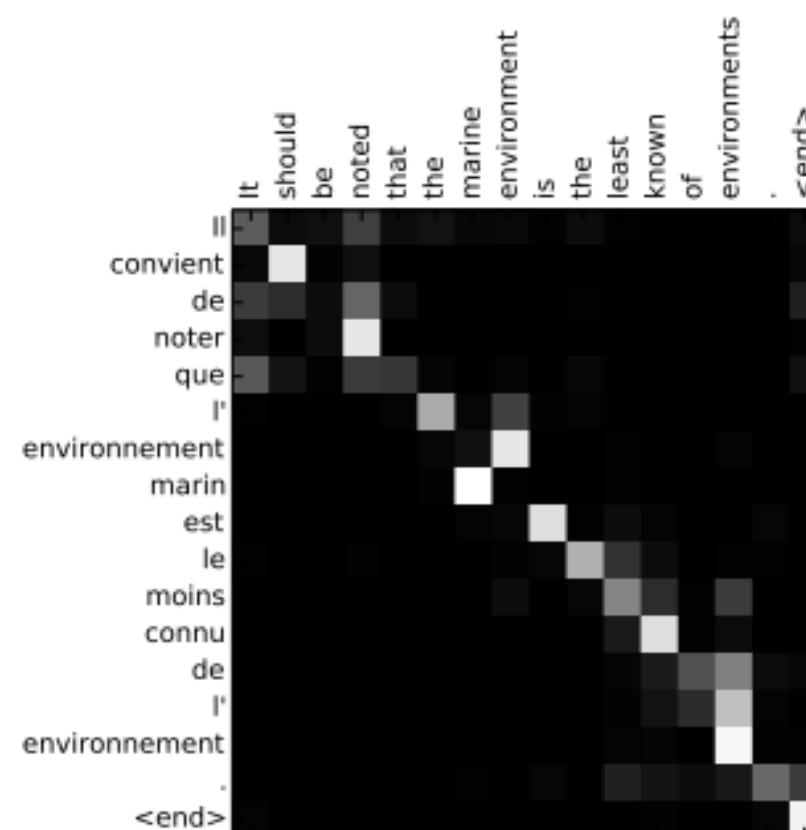


# ATTENTION

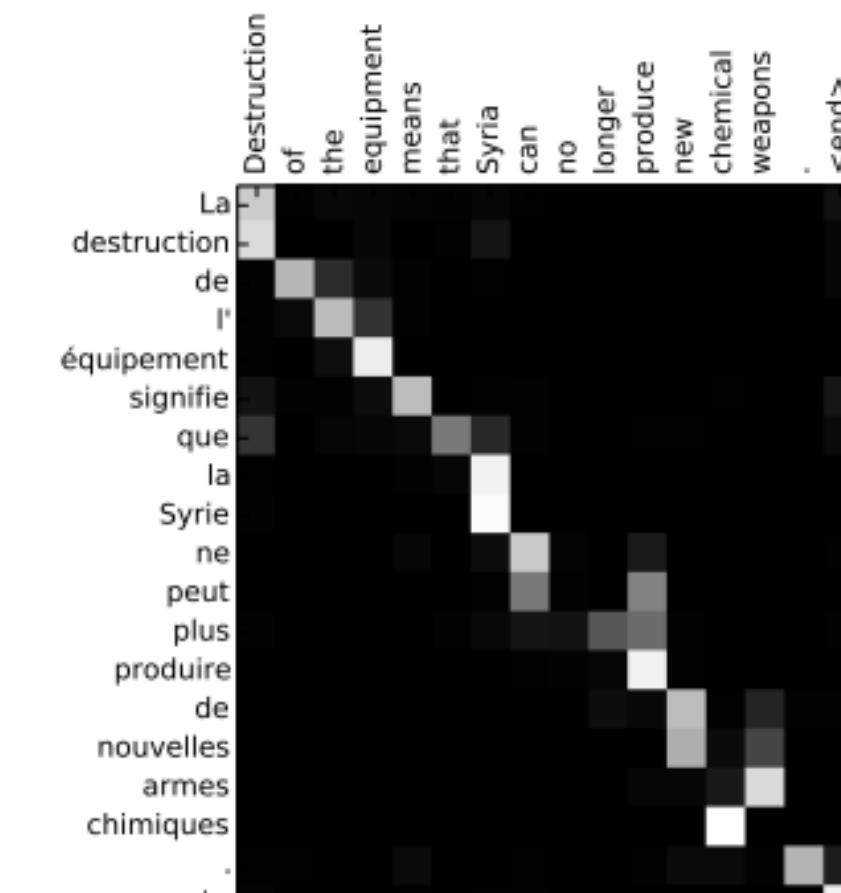
## The mechanism



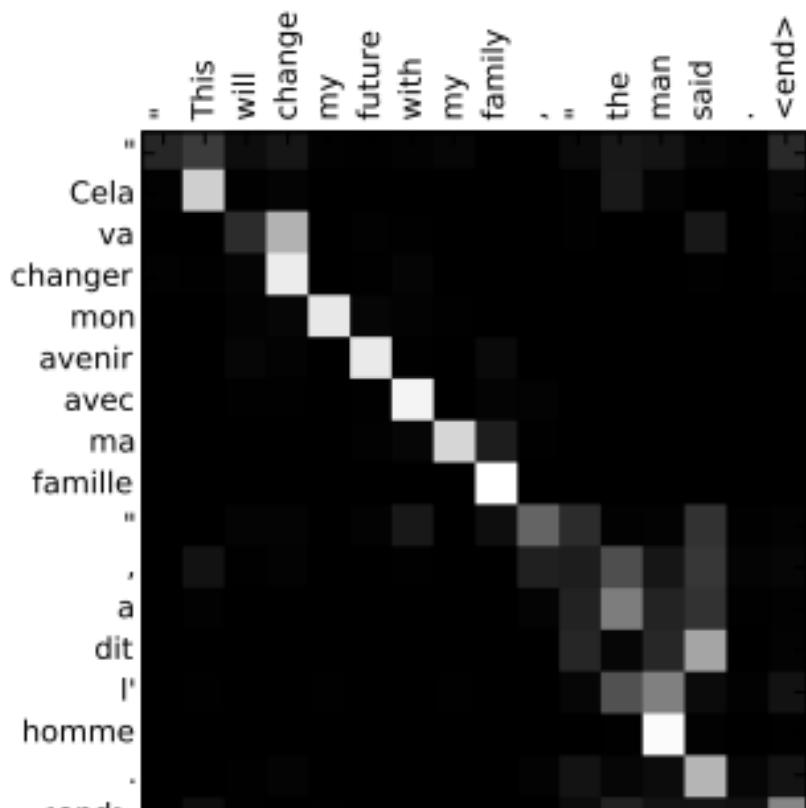
(a)



(b)



(c)

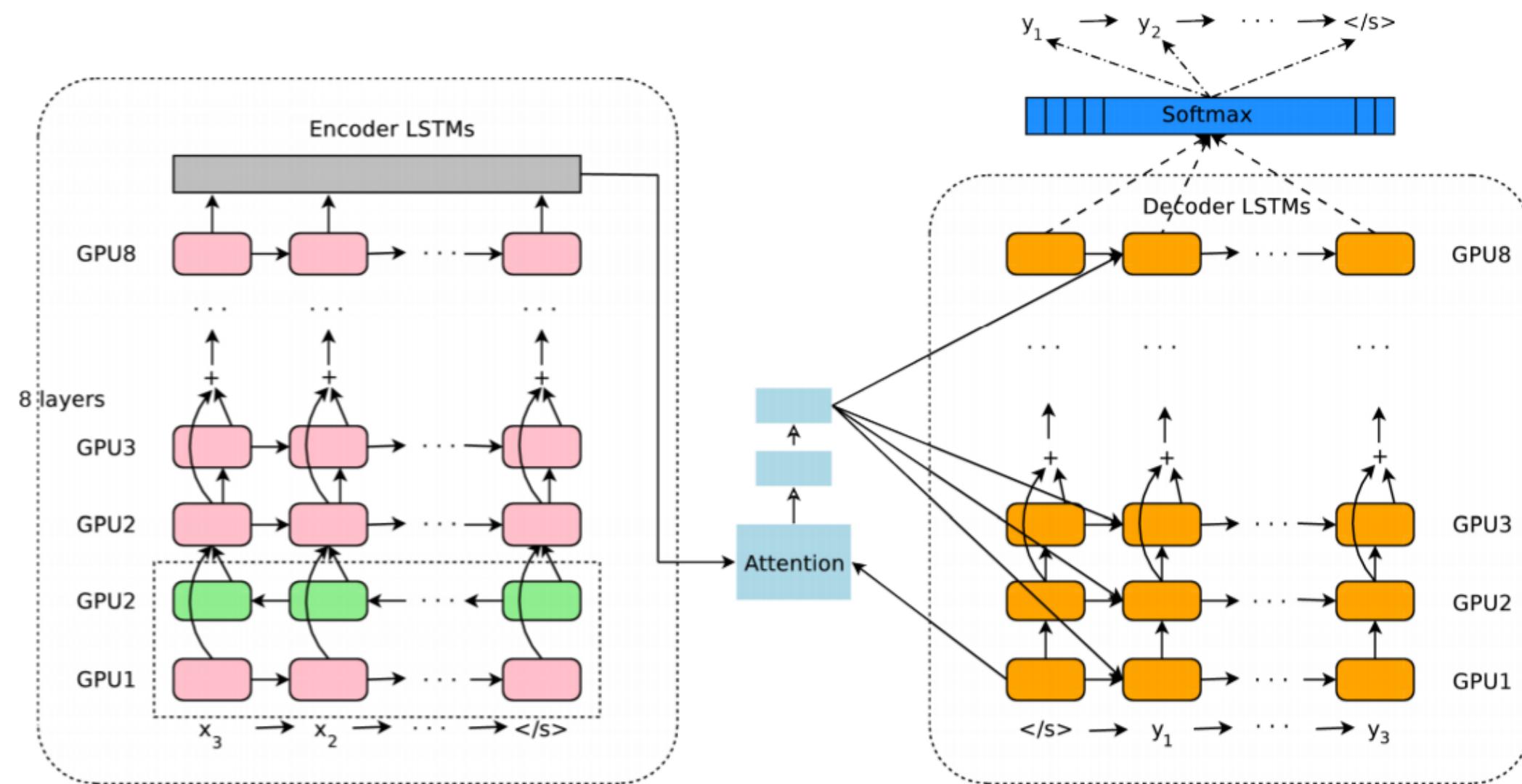


(d)

Figure 3: Four sample alignments found by RNNsearch-50. The x-axis and y-axis of each plot correspond to the words in the source sentence (English) and the generated translation (French), respectively. Each pixel shows the weight  $\alpha_{ij}$  of the annotation of the  $j$ -th source word for the  $i$ -th target word (see Eq. (6)), in grayscale (0: black, 1: white). (a) an arbitrary sentence. (b-d) three randomly selected samples among the sentences without any unknown words and of length between 10 and 20 words from the test set.

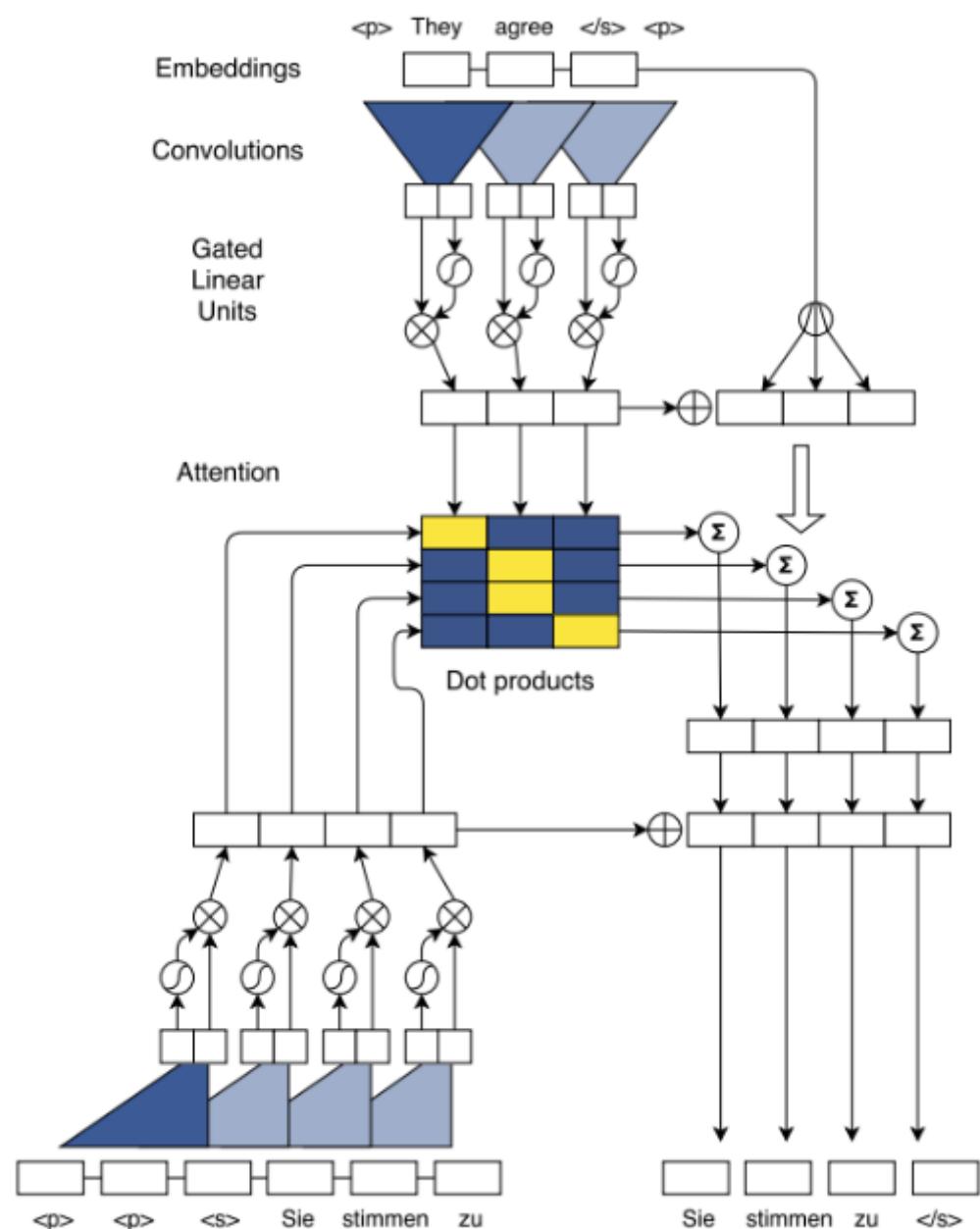
# ATTENTION

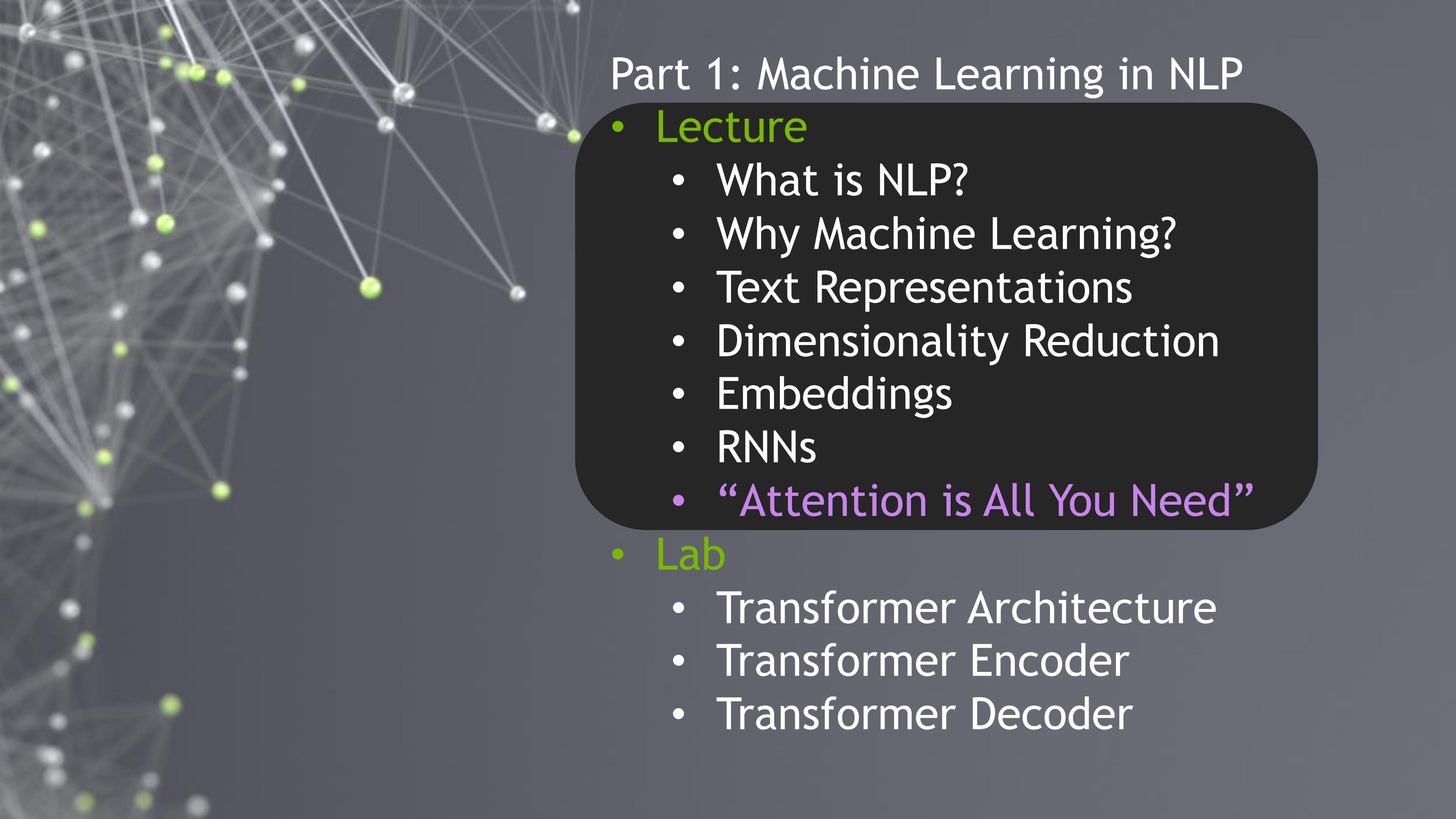
## Examples



# ATTENTION

## Examples





# Part 1: Machine Learning in NLP

- **Lecture**
  - What is NLP?
  - Why Machine Learning?
  - Text Representations
  - Dimensionality Reduction
  - Embeddings
  - RNNs
  - “Attention is All You Need”
- **Lab**
  - Transformer Architecture
  - Transformer Encoder
  - Transformer Decoder

# ATTENTION IS ALL YOU NEED

## Design

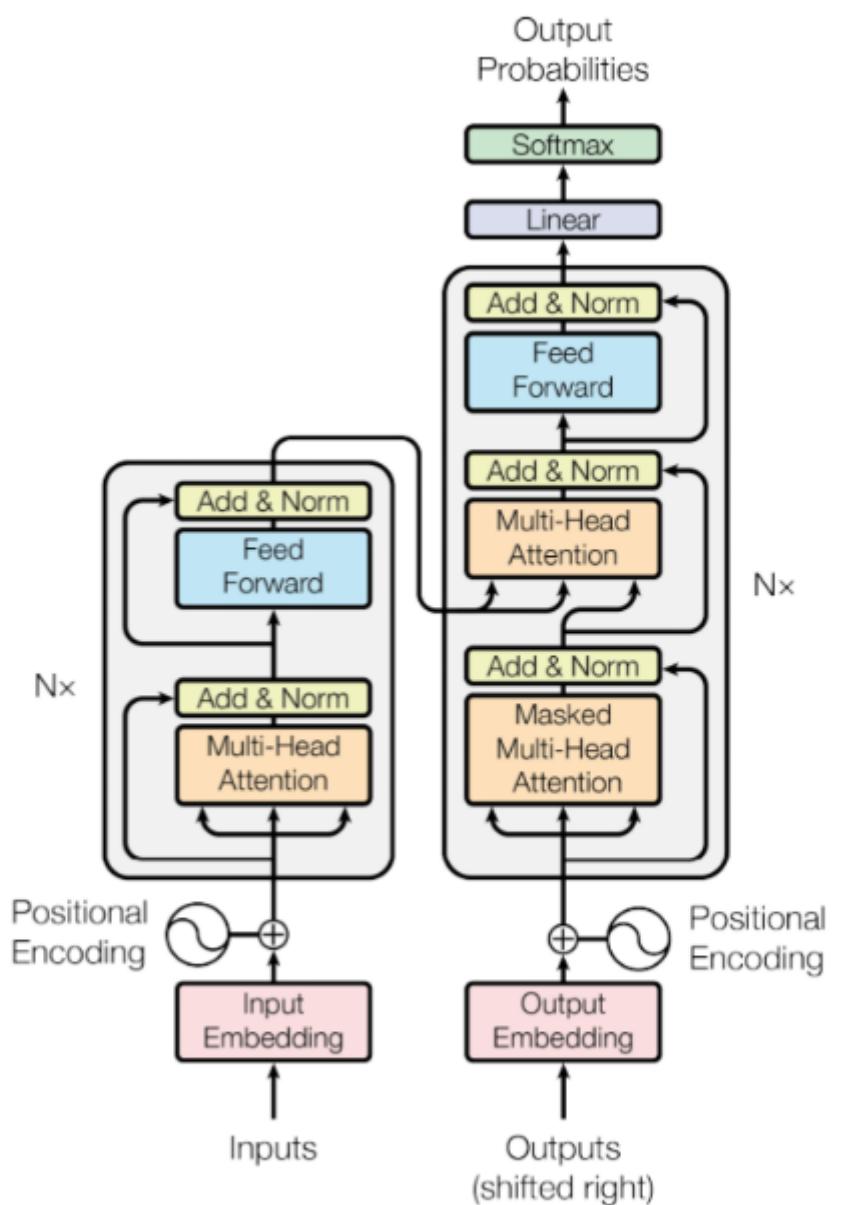


Figure 1: The Transformer - model architecture.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

# ATTENTION IS ALL YOU NEED

## Design

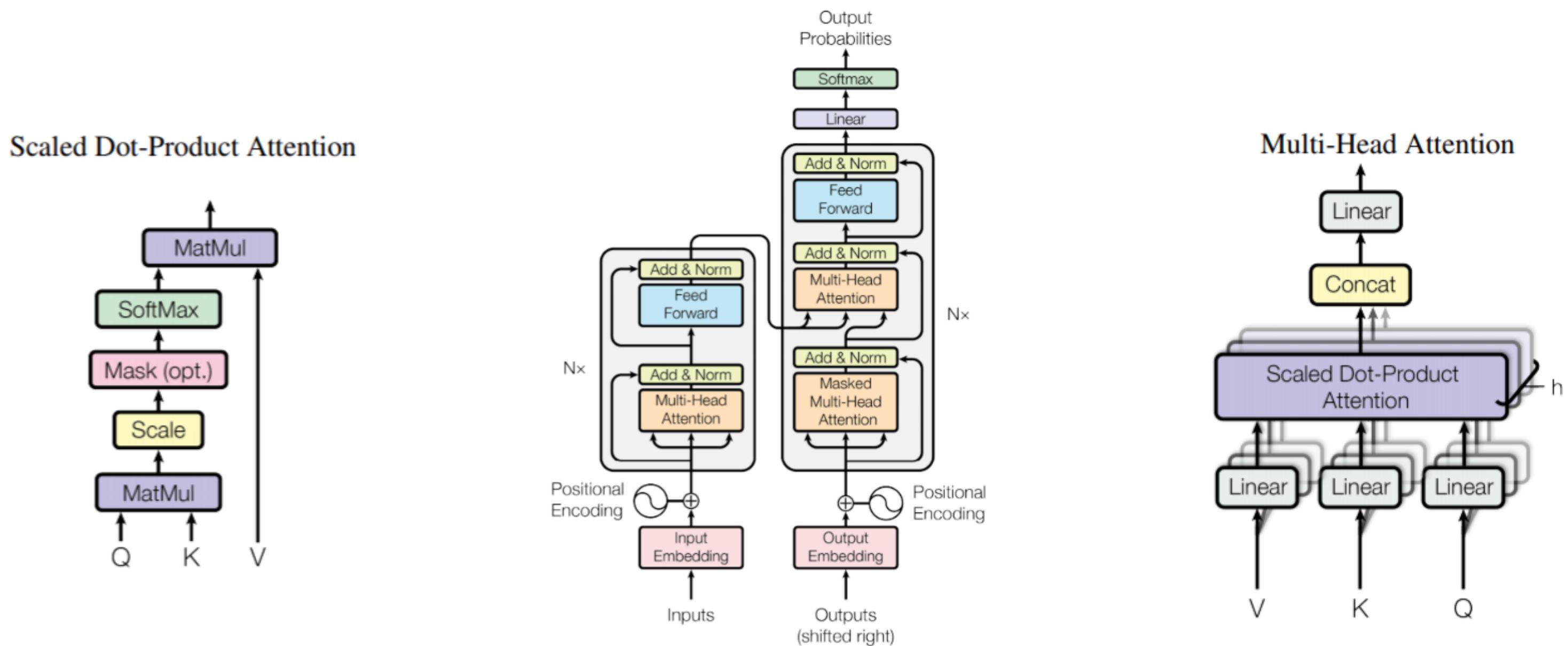


Figure 1: The Transformer - model architecture.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

# ATTENTION IS ALL YOU NEED

## Not a breakthrough in itself

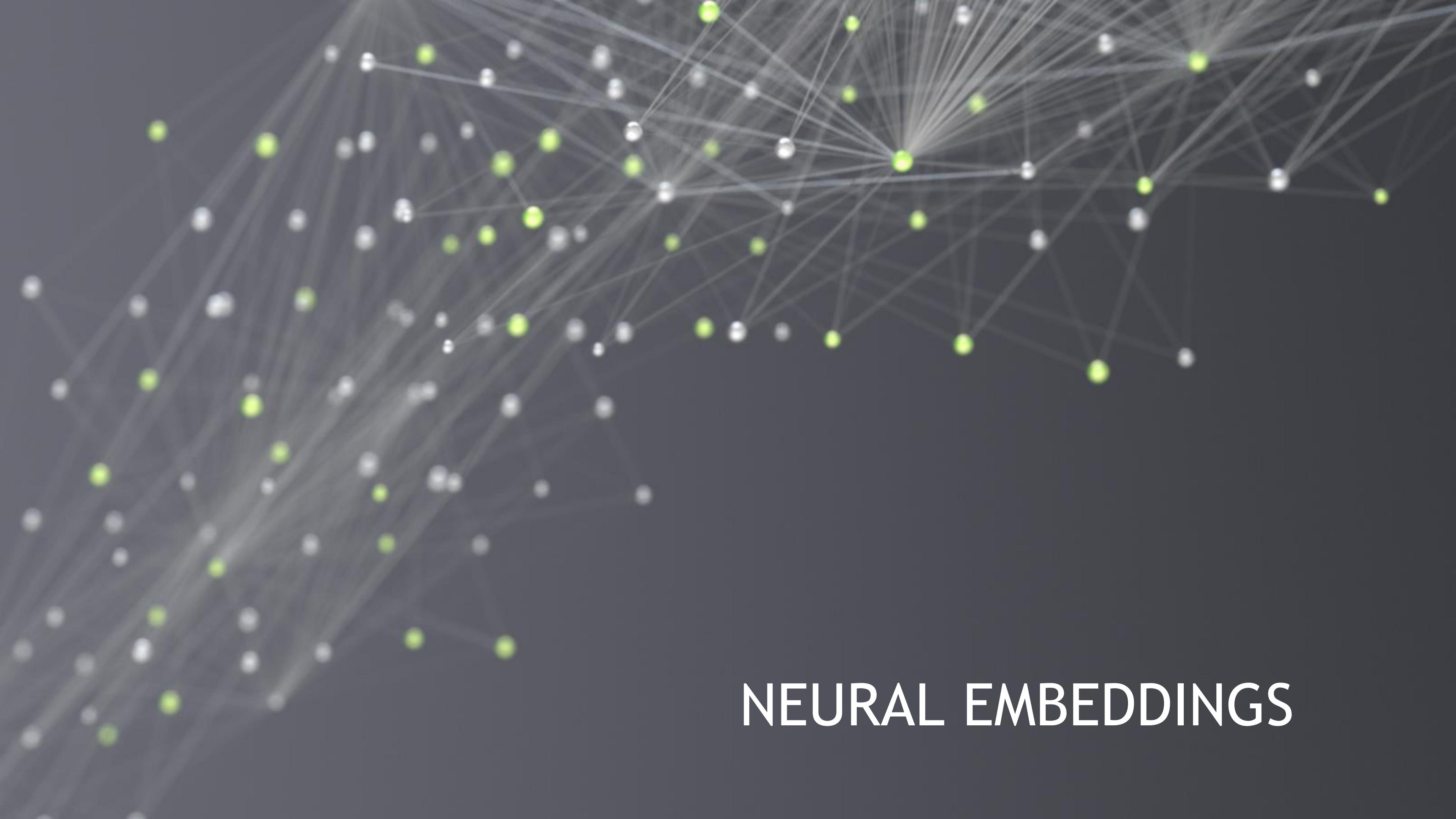
Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	<b>41.29</b>	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	<b><math>3.3 \cdot 10^{18}</math></b>	
Transformer (big)	<b>28.4</b>	<b>41.0</b>		$2.3 \cdot 10^{19}$

# ATTENTION IS ALL YOU NEED

But ...

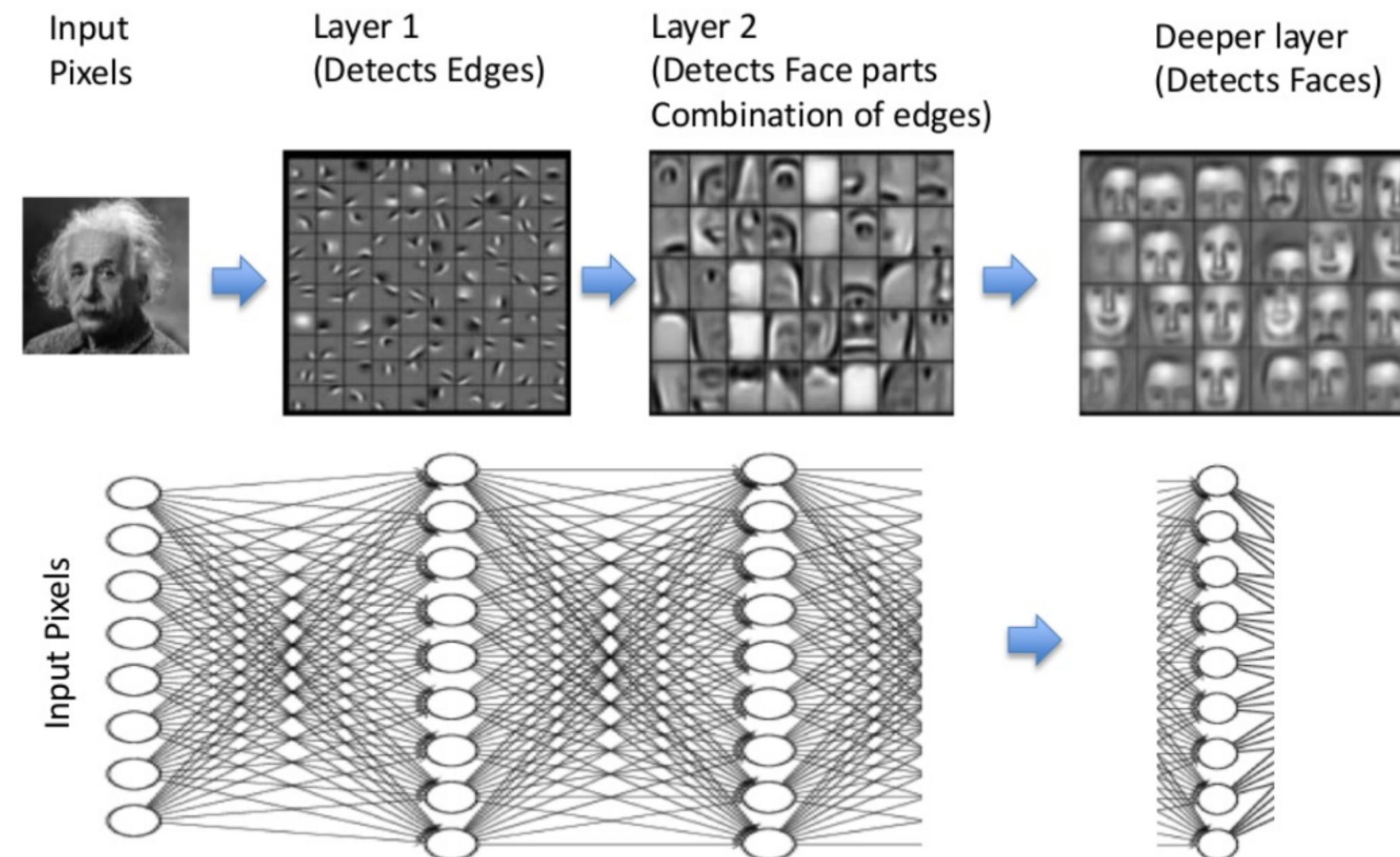
*“... the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers.”*

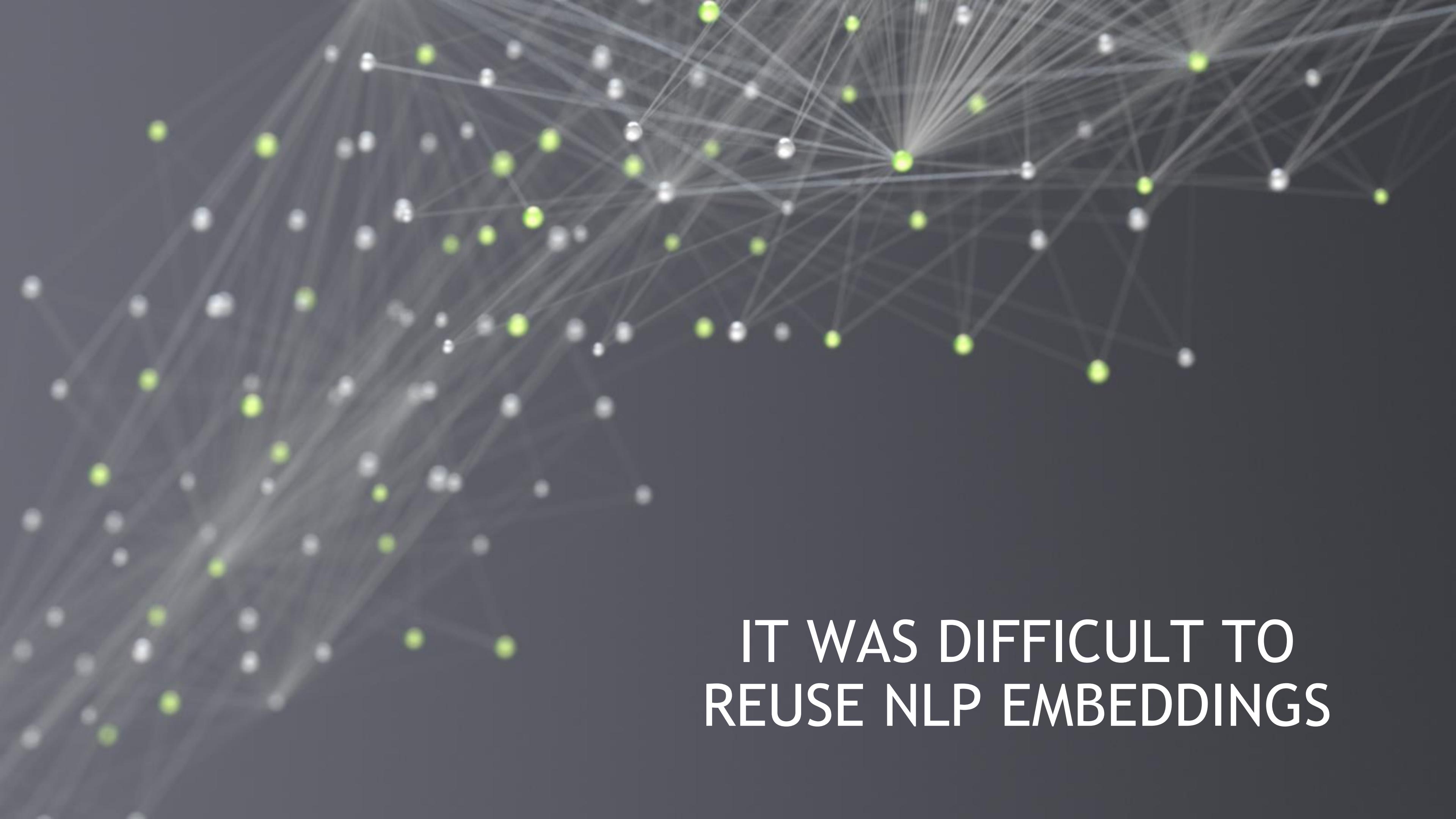


NEURAL EMBEDDINGS

# FEATURE REUSE

The opportunity





IT WAS DIFFICULT TO  
REUSE NLP EMBEDDINGS

# SEMI-SUPERVISED SEQUENCE LEARNING

## More complex representations

We present two approaches that use unlabeled data to improve sequence learning with recurrent networks. The first approach is to predict what comes next in a sequence, which is a conventional language model in natural language processing. The second approach is to use a sequence autoencoder, which reads the input sequence into a vector and predicts the input sequence again. These two algorithms can be used as a “pretraining” step for a later supervised sequence learning algorithm. In other words, the parameters obtained from the unsupervised step can be used as a starting point for other supervised training models. In our experiments, we find that long short term memory recurrent networks after being pretrained with the two approaches are more stable and generalize better. With pretraining, we are able to train long short term memory recurrent networks up to a few hundred timesteps, thereby achieving strong performance in many text classification tasks, such as IMDB, DBpedia and 20 Newsgroups.

# SEMI-SUPERVISED SEQUENCE LEARNING

## More complex representations

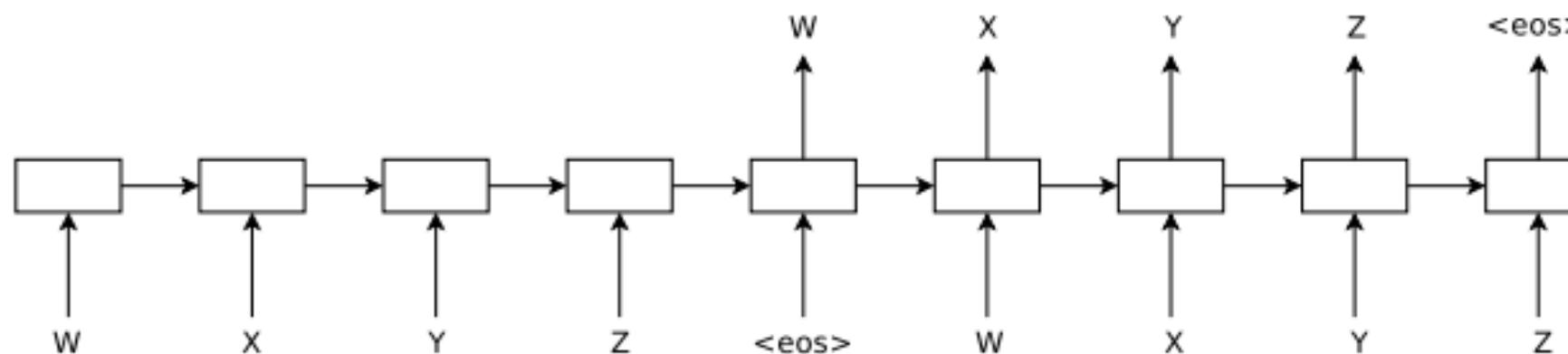


Figure 1: The sequence autoencoder for the sequence “WXYZ”. The sequence autoencoder uses a recurrent network to read the input sequence into the hidden state, which can then be used to reconstruct the original sequence.

# SEMI-SUPERVISED SEQUENCE LEARNING

## More complex representations

After training the recurrent language model or the sequence autoencoder for roughly 500K steps with a batch size of 128, we use both the word embedding parameters and the LSTM weights to initialize the LSTM for the supervised task. We then train on that task while fine tuning both the embedding parameters and the weights and use early stopping when the validation error starts to increase. We choose the dropout parameters based on a validation set.

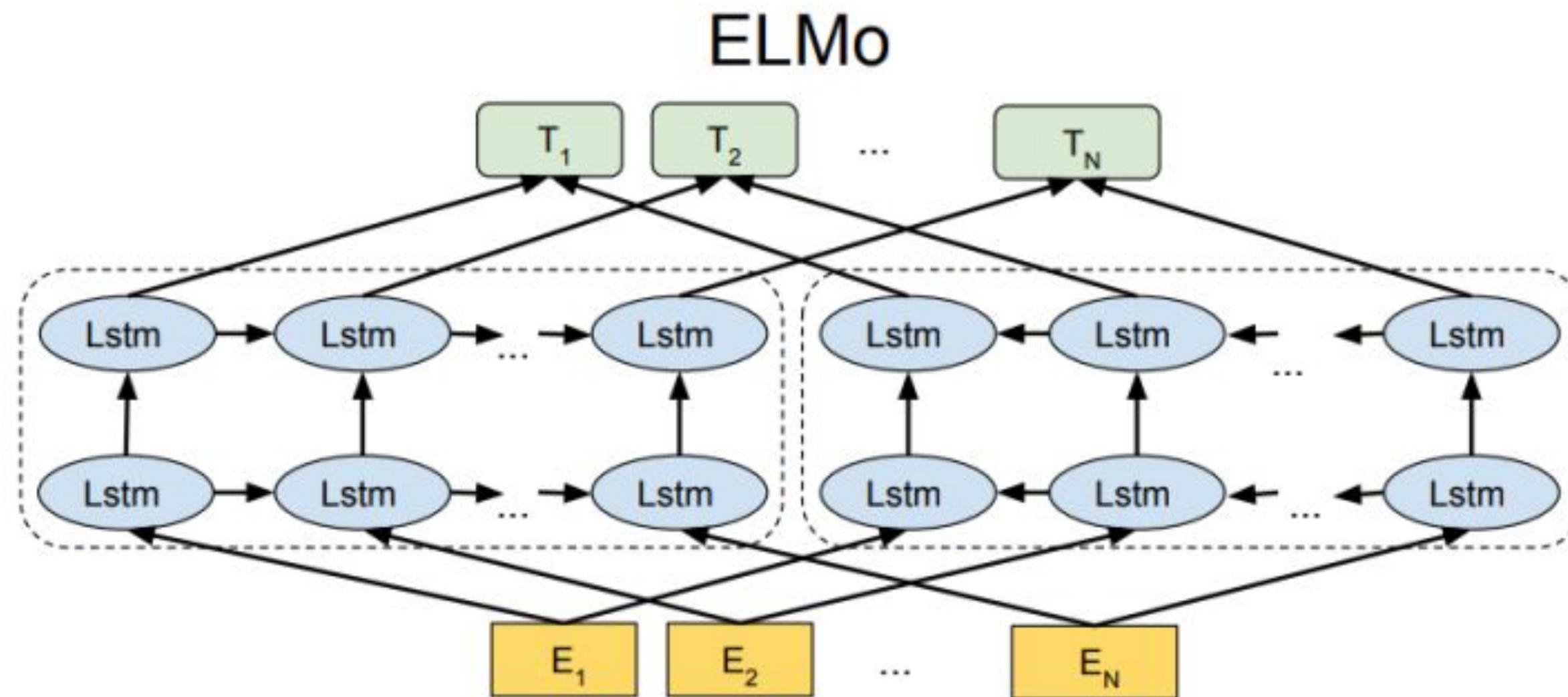
Using SA-LSTMs, we are able to match or surpass reported results for all datasets. It is important to emphasize that previous best results come from various different methods. So it is significant that one method achieves strong results for all datasets, presumably because such a method can be used as a general model for any similar task. A summary of results in the experiments are shown in Table 1. More details of the experiments are as follows.

Table 1: A summary of the error rates of SA-LSTMs and previous best reported results.

Dataset	SA-LSTM	Previous best result
IMDB	7.24%	7.42%
Rotten Tomatoes	16.7%	18.5%
20 Newsgroups	15.6%	17.1%
DBpedia	1.19%	1.74%

# ELMO

## Embeddings for Language Models



# ELMO

## Embeddings for Language Models

TASK	PREVIOUS SOTA	OUR BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17
SRL	He et al. (2017)	81.7	81.4	84.6
Coref	Lee et al. (2017)	67.2	67.2	70.4
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5

Table 1: Test set comparison of ELMo enhanced neural models with state-of-the-art single model baselines across six benchmark NLP tasks. The performance metric varies across tasks – accuracy for SNLI and SST-5;  $F_1$  for SQuAD, SRL and NER; average  $F_1$  for Coref. Due to the small test sizes for NER and SST-5, we report the mean and standard deviation across five runs with different random seeds. The “increase” column lists both the absolute and relative improvements over our baseline.

# ULM-FIT

## Universal Language Model Fine-Tuning for Text Classification

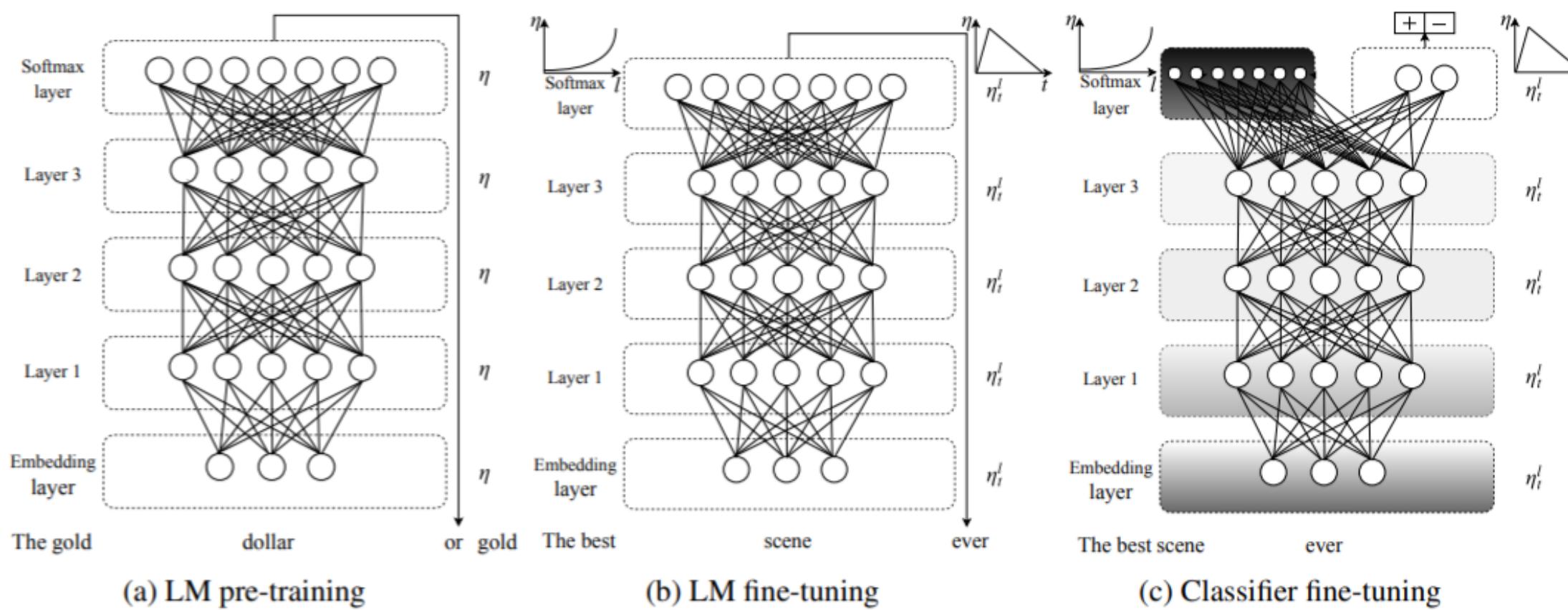


Figure 1: ULMFiT consists of three stages: a) The LM is trained on a general-domain corpus to capture general features of the language in different layers. b) The full LM is fine-tuned on target task data using discriminative fine-tuning ('*Discr*') and slanted triangular learning rates (STLR) to learn task-specific features. c) The classifier is fine-tuned on the target task using gradual unfreezing, '*Discr*', and STLR to preserve low-level representations and adapt high-level ones (shaded: unfreezing stages; black: frozen).

# DIFFICULTY

Not trivial to use and not universally applicable

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.



Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.





THIS CREATED A  
FOUNDATION FOR THE  
NEW NLP MODELS  
(DISCUSSED IN THE NEXT CLASS)



THE LAB

# ATTENTION IS ALL YOU NEED

Deep dive into the transformer design

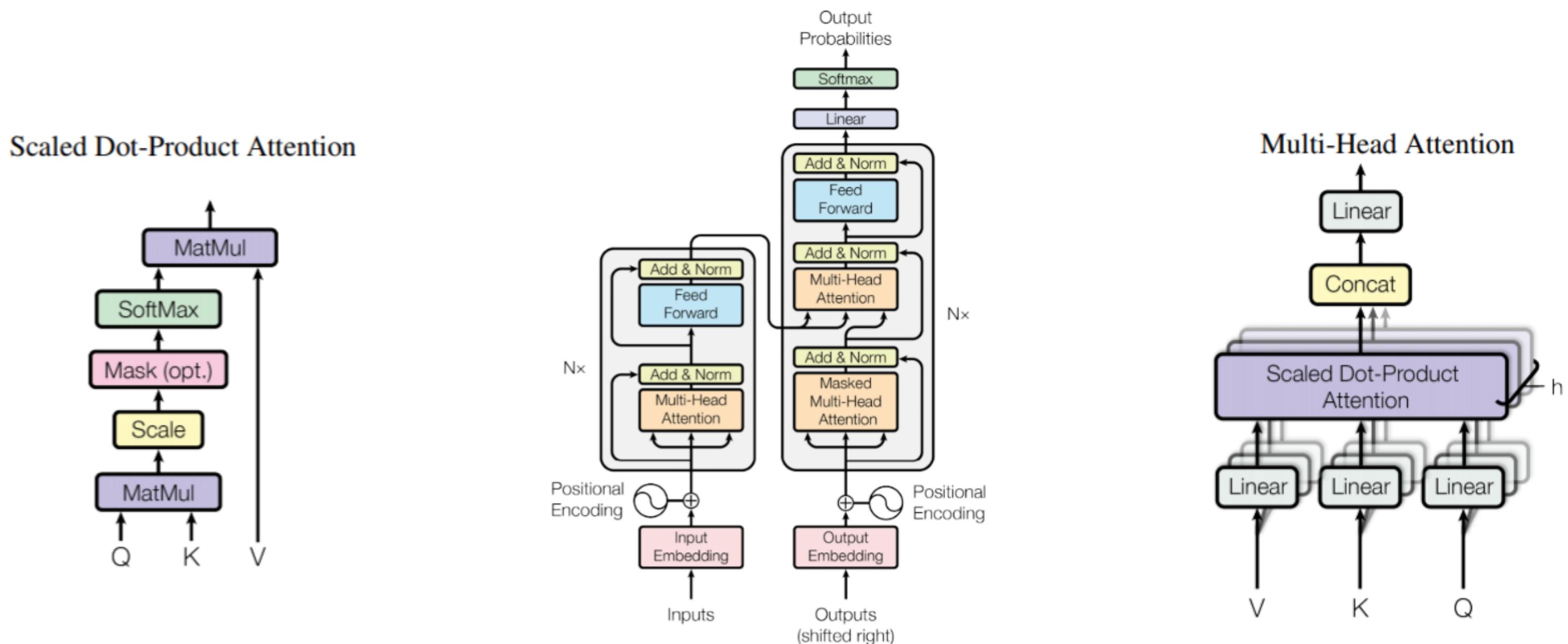


Figure 1: The Transformer - model architecture.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).



IN THE NEXT CLASS...

# SELF-SUPERVISION, BERT, AND BEYOND

Why did models start to work well? What does the future hold?

?